

THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: **Naoki OGUCHI**

Filed : **Concurrently herewith**

For : **PACKET PROCESSING DEVICE**

Serial No. : **Concurrently herewith**

April 19, 2000

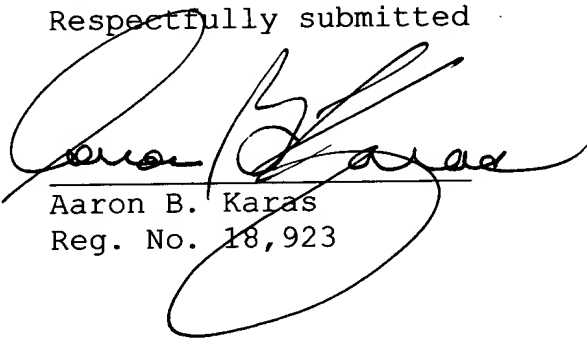
Assistant Commissioner of Patents
Washington, D.C. 20231

SUBMISSION OF PRIORITY DOCUMENT

S I R:

Attached herewith is Japanese patent application No.
11-137805 of May 18, 1999 whose priority has been claimed in the
present application.

Respectfully submitted


Aaron B. Karas
Reg. No. 18,923

HELFGOTT & KARAS, P.C.
60th FLOOR
EMPIRE STATE BUILDING
NEW YORK, NY 10118
DOCKET NO.: FUJZ17.260
LHH:priority

Filed Via Express Mail
Rec. No.: EL522392015US
On: April 19, 2000
By: Lydia Gonzalez
Any fee due with this paper, not fully
Covered by an enclosed check, may be
Charged on Deposit Acct. No. 08-1634

#2
H. McGee
7-12-00
jc525 U.S. PTO
09/552135
04/19/00

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

JCS25 U.S. PTO
09/552135
04/19/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 5月18日

出 願 番 号

Application Number:

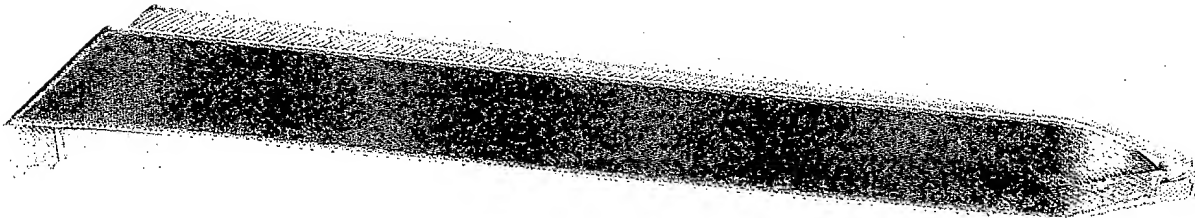
平成11年特許願第137805号

出 願 人

Applicant (s):

富士通株式会社

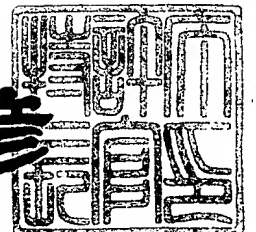
CERTIFIED COPY OF
PRIORITY DOCUMENT



2000年 2月18日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3007330

【書類名】 特許願

【整理番号】 9804689

【提出日】 平成11年 5月18日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/28
H04L 29/06
H04L 29/10

【発明の名称】 パケット処理装置

【請求項の数】 13

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 小口 直樹

【特許出願人】
【識別番号】 000005223
【氏名又は名称】 富士通株式会社

【代理人】
【識別番号】 100090011
【弁理士】
【氏名又は名称】 茂泉 修司

【手数料の表示】
【予納台帳番号】 023858
【納付金額】 21,000円

【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9704680

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 パケット処理装置

【特許請求の範囲】

【請求項 1】

階層化された通信プロトコルを処理するパケット処理装置において、
上位層の受信バッファと、
該受信バッファの空き容量を通知する受信バッファ空き容量通知部と、
該空き容量に基づき複数の受信パケットを 1 つのビッグパケットに再構成して
該受信バッファへ送信する再構成バッファ処理部と、
該ビッグパケットのサイズを該空き容量に基づいて決定する蓄積条件判定部と
、
を有することを特徴としたパケット処理装置。

【請求項 2】 請求項 1 において、

該受信バッファ空き容量通知部が、該上位層に含まれ、該空き容量を該蓄積条件判定部に通知することを特徴としたパケット処理装置。

【請求項 3】 請求項 1 において、

該受信バッファ空き容量通知部が、該上位層からの逆方向パケット内の情報に基づき該空き容量を検出する逆方向パケット内情報読出回路であることを特徴としたパケット処理装置。

【請求項 4】 請求項 2 又は 3 において、

該上位層が、トランスポート層であることを特徴としたパケット処理装置。

【請求項 5】 請求項 2 又は 3 において、

該上位層が、アプリケーション層であり、該ビッグパケットが、トランスポート層のバッファを介さずに、直接、該受信バッファに送信されることを特徴としたパケット処理装置。

【請求項 6】 請求項 1 において、

該受信パケットのコネクションを識別するコネクション識別回路をさらに有し

、
該再構成バッファ処理部が、該識別回路の識別情報に基づいてコネクション毎

に該ビッグパケットに再構成することを特徴としたパケット処理装置。

【請求項 7】請求項 1 において、

該ビッグパケットにチェックサムを付加するチェックサム計算回路を有することを特徴としたパケット処理装置。

【請求項 8】請求項 1 において、

該蓄積条件判定部は、所定時間が経過したとき、該ビッグパケットを該受信バッファに送信する旨の指示を該再構成バッファ処理部に与えるタイマを有することを特徴としたパケット処理装置。

【請求項 9】請求項 1 において、

該蓄積条件判定部は、該上位層からの確認応答パケットが発行されるサイズに達した時点で該ビッグパケットを該受信バッファに渡すための指示を該再構成バッファ処理部に与えることを特徴としたパケット処理装置。

【請求項 10】請求項 1 において、

該再構成バッファ処理部が、該ビッグパケットを、ヘッダを含む最初の該受信パケットと、それに続く受信パケットでヘッダを削除したものとで構成することを特徴としたパケット処理装置。

【請求項 11】請求項 1 において、

該受信パケットが非蓄積パケットであるとき、該再構成バッファ処理部に記憶せずに、直ちに、該受信バッファに送信する手段を有することを特徴としたパケット処理装置。

【請求項 12】請求項 1 において、

ネットワーク層のパケット転送機能をハード化した L3 スイッチであって、受信した自局宛の複数の受信パケットを該再構成バッファ処理部に送信するものを有することを特徴としたパケット処理装置。

【請求項 13】

請求項 1 に係るパケット処理装置を含んだ NIC 装置であって、

受信した自局宛の複数の受信パケットを該再構成バッファ処理部に送信することを特徴とした NIC 装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はパケット処理装置に関し、特に階層化された通信プロトコルを持つパケット処理装置に関するものである。

近年、階層型の構造を持つ通信プロトコル、例えばTCP/UDP/IPを持つパケット処理装置が、ルータ、ネットワーク層のパケット処理をハードウェアで行うレイヤ3スイッチ（以後、L3スイッチと略称する）、及びホストシステムのネットワークインタフェース(Network Interface Card:以後、NIC装置と略称する)等に用いられ普及している。

【0002】

また、近年の通信技術の進歩はめざましく、伝送速度はますます高速化されており、これにパケット処理装置は対応する必要がある。

【0003】

【従来の技術】

ルータ及びL3スイッチは、ネットワーク層でパケットを中継することが主な機能であるが、パケットを終端して、上位層に送り自局の通信もサポートする。また、NIC装置は、主にパケットを始終端して、上位層の送受信を実現するが、NIC装置を複数持つことにより、パケットの中継を行うことも可能である。パケット処理装置は、このような層間のパケット送受信処理を行うものである。

【0004】

以下に、こうしたパケット処理装置を含む機器についてより詳しく説明する。

ルータは、インターネット/イントラネットにおける(サブ)ネットワーク間中継装置として利用される機器である。サブネット間では、一般的にOSI7階層プロトコルモデルにおける第3層レベルのパケット中継を行う。通信プロトコルとしてTCP/IPを利用している場合は、ルータは第3層であるIPパケットを中継する。

【0005】

また、ルータは、パケットを始終端してルータ間の通信を行い、リモート制御をサポートする。

L3スイッチは、ルータと同様に、インターネット/イントラネットにおける(サブ

ネットワーク間中継装置として利用される機器であり、OSI7階層モデルにおける第3層レベルの packets 中継を行うが、ルータと異なり、IP中継処理をハードウェアで実現している。

【0006】

すなわち、L3スイッチは、ルータのネットワーク層の packets 転送機能をハードウェア化したものと言える。

また、ルータと同様に、L3スイッチは、packets を始終端してL3スイッチ間の通信を行い、リモート制御をサポートする。

【0007】

NIC装置は、主に通信の終始端点である端末のインタフェースとして使用され、通信端末上で稼働する通信アプリケーションソフトウェアと連動して、アプリケーションが packets を送受信するためのプロトコル処理の一部を担う。また、複数のNIC装置を実装することで通信ソフトウェアによって packets 中継を行うことも可能である。

【0008】

このような機器においては、プロトコル処理を高速化するために、受信した packets を効率的にハードウェアから上位層のオペレーティングシステム(以後、OSと略称することがある)に渡すための様々な工夫が行われてきた。

以下に、中継処理装置がIP packets を受信処理する際の従来のプロトコル処理方式を説明する。

(1)Zero-copy TCP方式

一般にアプリケーション実行用OSではOSカーネルが動作するメモリ空間と、ユーザプログラムが動作するメモリ空間を厳密に分けている。そのため、packets を受信し、TCPソケットバッファに到着したデータをOS上で動作する通信アプリケーションに渡すためにはカーネルメモリ空間からユーザメモリ空間へのメモリコピーを一定の手続きに従って行う必要がある。

【0009】

zero-copy TCP方式は、OSカーネルメモリ空間の一部をアプリケーションユーザ空間からもアクセス可能な機構を提供し、TCPの受信ソケットバッファをア

アプリケーションがアクセスできるようにする。

これにより、OS内プロトコル処理で処理時間の重いメモリコピーの必要が無くなり、OSプロトコル処理は、高速化する。

(2) I20方式

この方式は、メインプロセッサ上のOSで行われていたプロトコル処理等を、ネットワークインタフェース上のサブプロセッサで行うインタフェース仕様に基づいている。

【0 0 1 0】

すなわち、メインプロセッサ上のOSとは別のOSがサブプロセッサで動いており、このOS内で通信プロトコル処理を行い、処理後のデータをメインプロセッサOSに渡すようになっている。

これにより、メインプロセッサの負荷が軽減されると共に、プロトコル処理が高速化される。

(3) 大型ホスト計算機の通信制御装置方式

特開平4-292039号公報には、ホスト計算機にチャンネルで接続された通信処理装置を用いたホスト計算機間の通信方式が開示されている。

【0 0 1 1】

一般に、ホスト計算機－通信処理装置間チャンネルにおいては、ホスト計算機のトランスポート層が生成する大きなサイズのデータパケットで通信が行われる。

本方式では、通信処理装置間のサブネット区間は伝送路のMTUに合わせて送信側ホスト計算機が送信する大きなサイズのパケットを分割し、受信側の通信処理装置で再度、元の大きなサイズのパケットに組み立て直して受信側ホスト計算機に送信する。

【0 0 1 2】

これにより、ホスト計算機－通信処理装置間のチャンネル使用効率を高めることが可能となる。

(4) L3スイッチ方式

この方式で用いるL3スイッチは、上述したように、ネットワーク層のパケット転送をハードウェア化したものである。また、L3スイッチは既存のルータのネッ

トワーク層でのパケット転送機能をハードウェア化したものと考えられ、ルータの代わりに使用することが可能である。

【0013】

また、ルータには、ファイアウォール、プロキシサーバ等の高度な通信制御機能を持つものがあり、各通信セッションを通過させてよいかどうかの判断を行う。L3スイッチの登場と共に、L3スイッチに該高度な通信制御機能を搭載するアーキテクチャが登場している。

【0014】

図19は、上記のアーキテクチャを示しており、L3スイッチ81の上でOSカーネル20が動作し、このOSカーネル20の上にアプリケーション11が動作する。

L3スイッチ81は、受信パケット90を、受信インタフェース31、ルーティング処理回路83、SW部84、出力バッファ85、及び送信インタフェース46を経由して、スイッチングした後、送信パケット95として送出する（同図のルート(1)）。

【0015】

また、受信パケット90の一部は、受信インタフェース31からOSカーネル20のプロトコル処理部86に送られてフィルタリング等が為され、出力バッファ85に戻される（同図のルート(2)）。さらに、受信パケット90の一部は、ファイアウォール等のアプリケーション11に送られトラフィックの監視が行われるようになっている（同図のルート(3)）。

【0016】

こうした、ハードウェアファイアウォール機器では、通信セッションを識別するための最初の数個のパケット、あるいは、認証を行うためのパケットをソフトウェアで識別した後、通信セッションのフィルタ条件をハードウェアで認識できるようハードウェアの持つフィルタリングテーブルに記録し、以降のセッション認識はハードウェアで高速に行うというものである。

【0017】

この場合、フィルタリング処理をL3スイッチ81のハードウェアで処理することで高速化を図るが、セッション認識のための数個のパケットや、L3スイッチ管理のための通信セッション、あるいは複雑な認証機能は上位層でソフト処理されて

いる。

【0018】

図20は、図19に示したL3スイッチ81をより詳細に示しており、受信パケットを入力する受信インタフェース(RX)31、ルーティング処理部75、ヘッダ書換部76、入力側SW部77、共有メモリ78、出力側SW部79、及び送信パケット95を送出する送信インタフェース(TX)46が直列接続されている。

【0019】

また、共有メモリ78、CAM74、CPU71、及びメインメモリ72は、バス73を介して接続され、共有メモリ78は、さらに、バッファ管理回路37に接続され、CAM74は、ルーティング処理部75に接続されている。

動作において、ネットワークから受信した受信パケット（イーサネットフレーム）90は、受信インタフェース31の入力バッファ（FIFO）に蓄積される。

【0020】

L3スイッチでは、L2（階層2）のフレームでスイッチングを行うことと、L3（階層3）のIPパケットでIPルーティングをハードウェアで行うこと、が可能である。

L2スイッチ機能は、各ポート毎に流れるイーサフレームの送信元アドレスを学習しており、イーサネットアドレスと出力インタフェースとを対応付けた学習テーブルを作成する。

【0021】

L2スイッチ機能は、受信したイーサフレームをCRCチェックした後、宛先アドレスをキーとして学習テーブルを検索する。学習テーブルが自局の受信インタフェースのイーサネットアドレスである場合は、自局宛のフレームであるとして、受信処理を行う。

【0022】

先ず、IPパケットの宛先IPアドレスをキーとして、ハードウェアルーティングテーブル75を検索する。ルーティングテーブル75はCAM(Content Addressable Memory)と呼ばれるメモリ74に、OSが管理するルーティング情報を元にIPアドレスとポート或はイーサネットアドレスとの対応表が書き込んであり、IPアドレス

を入力することで出力サポートあるいは、イーサネットアドレスを出力として得ることができる。

【0023】

ルーティングテーブル75を検索し、自局宛のパケットではなく他のルータ或いはホストへ転送する必要がある場合は、イーサフレームヘッダの宛先アドレス及び送信元アドレスには、それぞれ、ルーティングテーブルより取得した次ホップのイーサネットアドレス、及び自局の出力ポートのイーサネットアドレスを付けて入力側SW部77へ送る。

【0024】

入力側SW部77は、フレームヘッダの宛先イーサネットアドレスを見て学習テーブルを参照し、出力すべきポートを決定する。その後、フレーム自体は共有メモリ78に格納し、ポート毎の出力順を管理するバッファ管理回路37の各インタフェース毎に設けられたキューにエントリを追加する。各キューのエントリ情報としては、フレームが納められている共有メモリの開始アドレスが記入される。

【0025】

出力側SW部79は、各ポート毎にバッファ管理回路37のキューから次に送信すべきフレームを取り出し、ポインタ情報を参照して共有メモリ78からフレームを取り出し送信インタフェース46に渡す。

送信インタフェース46は、パケットを伝送路に送るためにCSMA/CDアクセス方式によりデータを送信する。

【0026】

また、受信パケットが自局宛のIPパケットであった場合は、バッファ管理回路37にCPU71宛のキューを用意し自局宛にスイッチングを行う。自局宛のキューにフレームが到着するとCPU71上で動作するOSに対し割込を上げ、OSが共有メモリ78にアクセスすることでフレームを受信する。

【0027】

企業内ネットワークは、外部ネットワークに対し、機密保持、外部からの攻撃を避けるためにファイアウォールと呼ばれる装置を用いてインターネットに接続することで、企業内ネットワークの内部からはインターネットにアクセス可能だが

、外部からは企業内網にアクセスできない仕組みとなっている。

【0028】

図21は、企業内サブネット63と、インターネット65をファイアウォールを介して接続する最も簡単な例を示している。

企業内サブネット63に接続されたホスト61は、プロキシサーバ・ルータ（L3スイッチ）64を経由して、外部のインターネット65に接続されたホスト62に接続されている。

【0029】

パケットは、ルータでパケットフィルタリングが行われた後、ファイアウォールでトラフィック監視がなされる。

同図(2)に示すように、一般にルータ67は専用の装置を用いるが、ファイアウォールはソフト製品である場合が多く、ワークステーション等の汎用コンピュータ上で動作させる場合が多い。

【0030】

すなわち、ルータ67、及びファイアウォール66をそれぞれ異なる装置を用いてシステムを構築してきた。

ところが、近年、IPパケットの転送をハードウェア化したL3スイッチ（ルータ）は、ネットワーク層のパケットフォワーディング処理を高速化することで、ルータの転送能力を飛躍的に向上させた。

【0031】

このL3スイッチは、ハードウェアによるパケットのフィルタリングも可能であるため、ファイアウォール製品の一部フィルタリング処理をL3スイッチハードウェアで行う装置も登場してきた。

こうした製品は、ファイアウォールソフトウェアがOS内部で行うよう設定するフィルター条件等をハードウェアで行うフィルタ条件に置き換え、L3スイッチハードウェアに設定することで、ハードウェアでのフィルタリングを可能としている。

(5) OSへの割込数を減らした改良型NIC装置方式

コンピュータシステムに搭載されるNIC装置においては、ハードウェアで受信

したパケットをOSに渡す際、OSへの割込数を減らすことでOSのオーバヘッドを回避することを狙った製品がある。

【0032】

一般に、NIC装置がパケットを受信すると、OSに対し外部割込をあげる。割り込まれたOSはNIC装置上の受信バッファからOSの管理するメモリ空間にパケットをコピーする。この場合、NIC装置がパケットを受信する毎にOSは割込を受け、パケット処理をするためOSでのオーバヘッドが大きかった。

【0033】

そこで、改良型NIC装置方式では、パケット受信した際、複数パケットを受信した後、OSに割込を掛け、OSは各パケットをOSのメモリ空間にコピーするという技術を用いて、OSへの割込を減らすことによりOS内における処理オーバヘッドを減らしている。

【0034】

【発明が解決しようとする課題】

以下に、このような（従来のパケット処理装置を搭載する）装置の課題について述べる。

(1)Zero-copy TCP方式は、OSカーネルアドレス空間とユーザアドレス空間とでデータをコピーするオーバヘッドを無くすが、OSがハードウェアでパケットを受信する動作は、従来のソフト処理と同じあり、ソフト処理するパケットが増加するに伴い、ハードウェアのパケット中継を妨げてしまう。

【0035】

(2)I20方式は、上位層OSから見たパケット受信は、OSへの割込回数が減っている点、受信するパケットのサイズがアプリケーション層のデータ単位にまで大きくなっている。しかしながら、ネットワークインタフェース内の下位層OSは通常のパケット受信を行っているので、ハードウェアのパケット中継という観点では、受信パケット数が増加するとハードウェア処理をブロッキングする。

【0036】

(3)大型ホスト計算機の通信制御装置方式

今日の通信ホストは、様々なLANメディアに対応するデータリンク層を含む通

信プロトコルを実装しており、特開平4-292039号公報に示すように、もともと大きなトランスポート層のパケットを、細かく分割して送信し、受信側ホストで再び大きなパケットに再構成するようなことは行なわない。

【0037】

(4)L3スイッチ方式では、OSが、パケット受信毎に上がってきた割込によってコンテキストを切り替えた後、ハードウェアの持つ受信バッファメモリにアクセスする。このL3スイッチでは、ネットワーク層パケット転送処理を高速なハードウェアで行っているが、OSからの該メモリアクセスにより、ハードウェアのパケット転送にブロッキングを生じ、ルーティング性能に影響を与える。

【0038】

(5)OSへの割込数を減らした改良型NIC装置方式では、ソフトウェアの割込処理は減るが、OSが、受信したパケットの処理を各パケット毎に行うため、OSのオーバヘッドは大きくなってしまう。

また、L3スイッチを実装したファイアウォール装置は、現状で以下のような問題点がある。

【0039】

(1)L3スイッチのハードウェアで実現可能なフィルタリング、ポリシー設定を前提に考えられているため、複雑なセッション認識などはソフト処理する必要がある。

(2)アプリケーションゲートウェイなどパケットの中身を見るようなアプリケーション経由のトラフィックは、L3スイッチでハードウェア処理を行えないため、十分な性能が発揮出来ない。

【0040】

従って上記の課題を考慮し、本発明は、階層化された通信プロトコルを処理するパケット処理装置において、各階層のパケット処理が他階層の処理に影響を与えることなく、パケット処理を高速化することを目的とする。

【0041】

【課題を解決するための手段】

上記の課題を解決するため、請求項1に係る本発明のパケット処理装置は、

上位層の受信バッファと、該受信バッファの空き容量を通知する受信バッファ空き容量通知部と、該空き容量に基づき複数の受信パケットを1つのビッグパケットに再構成して該受信バッファへ送信する再構成バッファ処理部と、該ビッグパケットのサイズを該空き容量に基づいて決定する蓄積条件判定部と、を有することを特徴としている。

【0042】

すなわち、図1の本発明に係るパケット処理装置10の原理(1)に示すように、下位層の再構成バッファ処理部301が、複数の受信パケット91を1つのビッグパケット92に再構成する。この際、蓄積条件判定部303が、受信バッファ空き容量通知部102からの受信バッファ101の空き容量93に基づいてビッグパケット92のサイズを決定する。

【0043】

そして、このサイズにビッグパケット92のサイズが達したとき、再構成バッファ処理部301は、ビッグパケット92を上位層プロトコル処理部100の受信バッファ101に送信することができる。

なお、空き容量通知部102は、後述するように、配置される層が上位層とは限らない。また、図示のコネクション識別回路302及びチェックサム計算回路304については後述する。

【0044】

これにより、上位層の受信バッファ101へのパケット到着数を削減することが可能となり、通信プロトコルにおけるパケット送信に必要な処理量を減らし、パケットを高速で伝送することが可能となると共に上位層におけるパケット処理量を減らすことが可能になる。

【0045】

また、請求項2に係る本発明においては、該受信バッファ空き容量通知部が、該上位層に含まれ、該空き容量を該蓄積条件判定部に通知することが可能である。

すなわち、上位層に配置された空き容量通知部102が、受信バッファ101の空き容量93を検出して、下位層の蓄積条件判定部303に通知するようにしてもよい。

【0046】

また、請求項3に係る本発明においては、該受信バッファ空き容量通知部を、該上位層からの逆方向パケット内の情報に基づき該空き容量を検出する逆方向パケット内情報読出回路としてもよい。

図2は、本発明の原理(2)を示しており、図1と異なる点は、上位層の空き容量通知部102の代わりに下位層に逆方向パケット内情報読出回路305が追加されていることである。

【0047】

すなわち、情報読出回路305を再構成バッファ処理部301と同じ層に配置し、上位層からの逆方向パケット94に含まれる受信バッファ101の空き容量情報を参照して受信バッファの空き容量93を知り、この空き容量93を蓄積条件判定部303に通知するようにしてもよい。

【0048】

また、請求項4に係る本発明においては、該上位層をトランスポート層とすることができる。

すなわち、受信バッファ101が、トランスポート層の受信バッファであり、トランスポート層の下位層に再構成バッファ処理部301を配置し、このバッファ処理部301で構成したビッグパケット92を受信バッファ101に送信することができる。

【0049】

また、請求項5に係る本発明においては、該上位層が、アプリケーション層であり、該ビッグパケットをトランスポート層のバッファでのフロー制御を介さずに、直接、該受信バッファに送信することが可能である。

図3は、本発明の原理(3)を示しており、図1と異なる点は、上位層プロトコル処理部100と下位層プロトコル処理部300の間に中間層プロトコル処理部200が有ることである。

【0050】

すなわち、下位層のトランスポート層に配置された再構成バッファ処理部301から、中間層のフロー制御を介さず、上位層のアプリケーション層の受信バッファ

ア101にビッグパケット内のデータを送信することが可能である。

これにより、中間層においてフロー制御を行わない通信プロトコルにおいて、パケットを高速で伝送することが可能となる。

【0051】

また、請求項6に係る本発明においては、該受信パケットのコネクションを識別するコネクション識別回路をさらに有し、該再構成バッファ処理部が、該識別回路の識別情報に基づいてコネクション毎に該ビッグパケットに再構成することが可能である。

【0052】

すなわち、図1に示したコネクション識別回路302が、受信パケットのコネクションを識別し、再構成バッファ処理部301が、同一コネクション毎にビッグパケット92を再構成することができる。

これにより、複数のコネクションに対応することが可能となる。

【0053】

また、請求項7に係る本発明においては、該ビッグパケットにチェックサムを付加するチェックサム計算回路を有することができる。

すなわち、図1に示したチェックサム計算回路304が、ビッグパケット92に対してチェックサムを計算して付加することが可能である。

【0054】

これにより、中間層プロトコル処理部でのチェックサムエラー検出において矛盾なくビッグパケットを受信可能となる。

また、請求項8に係る本発明においては、

該蓄積条件判定部は、所定時間が経過したとき、該ビッグパケットを該受信バッファに送信する旨の指示を該再構成バッファ処理部に与えるタイマを有することが可能である。

【0055】

すなわち、蓄積条件判定部303は、再構成バッファ処理部301がビッグパケット92を構成するときタイマをスタートする。そして、ビッグパケット92が所定のサイズに達する前にタイムアウトしたとき、これを受けた再構成バッファ処理部30

1はビッグパケット92を受信バッファ101に送信する。

【0056】

これにより、ビッグパケット92を構成する時間が長くなることによる伝送遅れや再送を回避することができる。

また、請求項9に係る本発明においては、該蓄積条件判定部は、該上位層からの確認応答パケットが発行されるサイズに達した時点で該ビッグパケットを該受信バッファに渡すための指示を該再構成バッファ処理部に与えることが可能である。

【0057】

すなわち、受信バッファ101が受信したパケット容量が、所定の容量以上になったとき、受信バッファ101に保持されたパケットはさらに上位層のバッファに送信され、確認応答（ACK）パケットが下位層に対して発行されるようになっているのが通常である。

【0058】

そこで、蓄積条件判定部303は、ACKパケットが発行されるサイズにビッグパケット92のサイズが達した時点で、受信バッファ101にビッグパケット92を送信する旨の指示を再構成バッファ処理部301に与える。

これにより、受信バッファ101に送信されたビッグパケット92は、即座に上位層のバッファに送信されると共にACKパケットが送信されることになり、パケットを高速で上位層に伝送することが可能となる。

【0059】

また、請求項10に係る本発明においては、該再構成バッファ処理部が、該ビッグパケットを、ヘッダを含む最初の該受信パケットと、それに続く受信パケットでヘッダを削除したものとで構成してもよい。

すなわち、再構成バッファ処理部は、最初の、ヘッダを削除しない受信パケットと、これに続く受信パケットのヘッダを削除したデータとでビッグパケットを構成してもよい。

【0060】

この結果、受信パケットと同じフォーマットのヘッダをビッグパケットの情報

を設定するヘッダに設定することが可能となり、上位層の変更無しに本パケット処理装置が採用できる。

また、ビッグパケットは、削除したヘッダ分だけデータ容量の少ない構成にすることになり、より高速でパケットを上位層に送信することが可能となる。

【 0 0 6 1 】

また、請求項11に係る本発明においては、該受信パケットが非蓄積パケットであるとき、該再構成バッファ処理部に記憶せずに、直ちに、該受信バッファに送信する手段を有することが可能である。

すなわち、緊急に処理すべきデータを含む受信パケット91を、再構成バッファ処理部301を経由せずに、直ちに、上位層の受信バッファ101に送信することが可能である。

【 0 0 6 2 】

また、請求項12に係る本発明においては、上記に加えてさらに、ネットワーク層のパケット転送機能をハード化したL3スイッチであって、受信した自局宛の複数の受信パケットを該再構成バッファ処理部に送信するものを有することができる。

【 0 0 6 3 】

すなわち、L3スイッチが、受信したパケットの中から自局宛の受信パケットのみを再構成バッファ処理部に送る。再構成バッファ処理部は、複数の受信パケットで該空き容量に基づいたサイズの1つのビッグパケットを再構成し、上位層の受信バッファに送ることが可能である。

【 0 0 6 4 】

これにより、L3スイッチのパケット中継動作をブロッキングすることなく自局宛の受信パケットを受信することが可能となり、パケット処理を高速化することができる。

さらに、請求項13に係る本発明においては、請求項1に係るパケット処理装置を含んだNIC装置であって、受信した自局宛の複数の受信パケットを該再構成バッファ処理部に送信することを特徴としている。

【 0 0 6 5 】

すなわち、NIC装置は、受信した自局宛の受信パケットをパケット処理装置に与える。パケット処理装置において、再構成バッファ処理部が、複数の受信パケットを該空き容量に基づいたサイズの1つのビッグパケットに再構成し、NIC装置の上位層の受信バッファに与える。

【0066】

これにより、NIC装置の上位層のOSは、自局宛受信パケットの処理回数を減少させることが可能となり、OSのオーバヘッドを無くすることができる。

【0067】

【発明の実施の形態】

図4は、本発明に係るパケット処理装置10の実施例(1)を示しており、このパケット処理装置10は、ネットワークと送信パケット95及び受信パケット90を送受信するインタフェースモジュール30、このモジュール30との間でパケットを送受信するOSカーネル20、及びこのOSカーネル20上で動作するアプリケーション層プロトコル処理部（アプリケーションプログラム、及びアプリケーションと同義である）11で構成されている。

【0068】

インタフェースモジュール30は、パケットを受信する受信インタフェース31、コネクション識別回路32、チェックサム計算回路33、ヘッダ削除回路34、再構成バッファ書込回路35、再構成バッファ36、バッファ管理回路37、バッファ条件判定回路38、タイマ39、受信キュー40、DMA転送回路41、コネクションキュー42、サービスリスト43、送信キュー44、送信バッファ45、及び送信パケット95を送信する送信インタフェース46で構成されている。

【0069】

なお、再構成バッファ書込回路35と再構成バッファ36とで図1に示した再構成バッファ処理部301を構成している。

OSカーネル20は、受信ソケットバッファ22を含むトランスポート層プロトコル処理部21、ネットワーク層プロトコル処理部24、及びドライバ（デバイスドライバ）25で構成されている。

【0070】

アプリケーション層プロトコル処理部11は、アプリケーションバッファ12を含んでいる。

以下に、実施例(1)の動作手順を図5に示したフローチャート図を参照して説明する。

【0071】

OSカーネル20上のプロトコル処理部11は、他のホストと仮想コネクションを作成し、通信を行う。ここでは他ホスト上のプロトコル処理部から、本プロトコル処理部11にコネクションが作成された場合を想定している。

トランスポート層プロトコル処理部21は、他ホストからのコネクションが作成されると、コネクション毎に受信ソケットバッファ22を作成する。

【0072】

そして、プロトコル処理部21は、ドライバ25に依頼し、インタフェースモジュール30のサービスリスト43にコネクションの識別情報及び受信ソケットバッファ22の空き容量93（図1参照）であるバッファリング閾値を書き込む。

なお、この動作は、初回だけでもよいし、頻繁に書き換えても良い。

【0073】

受信インタフェース31は、物理ネットワークから受信パケット90を受信し、コネクション識別回路32に送る（図5のステップS10）。

コネクション識別回路32は、管理するコネクション宛の受信パケット90が、遅延を許さない、例えばACKパケット及び緊急フラグ＝“オン”であるパケット等の非蓄積パケットである場合、非蓄積パケットを蓄積せずにOSカーネル20に送信すると共に、後述する再構成バッファのビッグパケットもOSカーネル20に直ちに送信する（同S11のYES）。

【0074】

コネクション識別回路32は、受信パケット90のネットワーク層ヘッダ及びデータリンク層ヘッダのコネクション識別情報をキーとしてサービスリストを検索し、自局に取り込みたいコネクションの受信パケット90であるか否かを識別する（同S12, S13）。

【0075】

該コネクション識別情報がサービスリスト43に無い場合、自局宛でないと判断し、通常のIPパケットとして転送処理を実行する（同S14）。

サービスリスト43でヒットした場合、チェックサム計算回路33は、受信パケットに含まれるトランスポート層のデータパケットが正常であることを調べるためにチェックサムを計算する（同S15）。

【 0 0 7 6 】

ヘッダ削除回路34は、受信パケット90が、蓄積を開始した先頭の受信パケットであるか否かを判定する（同S16）。

先頭のパケットである場合、ヘッダ削除回路34は、ネットワーク層及びデータリンク層のヘッダを付けたまま再構成バッファ書込回路35に渡す。書込回路35は、バッファ管理回路37に問い合わせることにより、再構成バッファ36に書き込みを開始する位置を取得し（同S17）、再構成バッファ36に書き込みを行う（同S18）。

【 0 0 7 7 】

さらに、書込回路35は、サービスリスト43を参照し、ポインタが指すメモリ上のコネクションキュー42に再構成バッファ36の書込アドレス（ビッグパケット開始アドレス）を書き込んだ後（同S19）、タイマ39を一定時間後にタイムアウトするようにセットする（同S20）。

【 0 0 7 8 】

先頭パケットでない場合、ヘッダ削除回路34は、受信パケット90のネットワーク層及びデータリンク層のヘッダを削除して書込回路35に渡す（同S21）。書込回路35は、ステップS17～S19と同じ動作を実行して、ビッグパケット92（図1参照）を構成して行く（同S22～S24）。

【 0 0 7 9 】

また、書込回路35は、コネクションキュー42で管理する受信パケット数及びビッグパケット長の情報を書き込む。

この結果、バッファ条件判定回路38は、サービスリスト43に記入されたバッファリング閾値93と、コネクションキュー42に記入されたビッグパケット長とを比較する。

【0080】

バッファリング閾値を越えた場合（同S25）、判定回路38は、再構成バッファ内のビッグパケットのビッグパケット開始アドレスをコネクションキュー42から取得し、これを受信キューに書き込み、コネクションキュー42はデキューしておく。また、OSカーネル20に対し、ビッグパケットを受信したことを知らせる割込信号を発行する（同S27）。

【0081】

受信パケット90が到着せず、ビッグパケット長が、バッファリング閾値を越えずにタイムアウトした場合（同S26）、これをタイマ39は、バッファ条件判定回路に通知する。バッファ条件判定回路38は、再構成バッファ内のビッグパケット開始アドレスを受信キュー40に書き込み、コネクションキュー42はクリアすることでデキューし、OSに対し、パケットを受信したことを知らせる割込を発行する（同S27）。

【0082】

割込を受けたOSカーネル20において、ドライバ25がOSメモリ空間内にビッグパケットを管理するための構造体を作成し、実際にメインメモリ（図示せず）上にビッグパケットをコピーするためのアドレス位置を決定する。

DMA転送回路41は、メインメモリに再構成バッファ36内で再構成されたビッグパケットをコピーし、コピー完了の割込をネットワーク層プロトコル処理部24に上げる。

【0083】

プロトコル処理部24は、受信したビッグパケットのヘッダ処理を行い、トランスポート層プロトコル処理部21に渡す。

プロトコル処理部21の受信ソケットバッファ22は、一般に、受信データを蓄積し、一定量蓄積した後、ACKパケットを返す動作を行う。

【0084】

ビッグパケットは、再構成バッファ36で既に確認応答を返す条件を満たすようなデータサイズで構成されるように設定されているので、受信ソケットバッファ22は、直ちに、ACKパケットを送信し、ビッグパケットをアプリケーションバッ

ファ12にコピーする。

【 0 0 8 5 】

この結果、OSカーネル20における受信パケットの処理負荷が軽減されることになる。

図6は、本発明のパケット処理装置10の実施例(2)を示している。このパケット処理装置10が実施例(1)と異なる点は、インタフェースモジュール30において、送信バッファ45と送信インタフェース46との間に、送信パケットをコネクションを識別するコネクション識別回路47、ウィンドウサイズ抽出回路48、受信バッファ長計算回路49、及び受信バッファ長書込回路50が直列に挿入されていることである。

【 0 0 8 6 】

実施例(2)の動作手順は、図5に示した実施例(1)のフローチャート図に組み合わせることができる図7のフローチャート図を追加した動作手順となる。

以下に、実施例(2)の動作手順を図7を参照して説明する。

本実施例(2)においても、実施例(1)と同様に、他のホストからOSカーネル20上のプロトコル処理部11に仮想コネクションが、作成された場合について述べる。

【 0 0 8 7 】

実施例(1)と同様に、コネクション毎に受信ソケットバッファ22が作成され、プロトコル処理部21は、ドライバ25に依頼し、サービスリスト43にコネクションの識別情報、及びソケットバッファの最大値をバッファリング閾値93として書き込む。

【 0 0 8 8 】

トランスポート層プロトコル処理部21は、ビッグパケットに対するACKパケット等の逆方向パケット94(図2参照、以後、送信パケット94と称する)を、OSカーネル20が動作しているメインメモリから送信バッファ45にDMA転送回路41を経由して転送し、送信パケット94は、送信順に送信キュー44にキューイングされる(図7のステップS30)。

【 0 0 8 9 】

送信パケット94は、送信キュー44に基づいて、送信バッファ45から取り出され、コネクション識別回路47に送られる。

コネクション識別回路47は、送信パケット94のネットワークヘッダ部及びトランスポート層ヘッダ部で示されるコネクションを、サービスリスト43を参照して検索する（同S31）。

【0090】

該当するコネクションがサービスリスト43に存在した場合は、送信パケット94をウィンドウサイズ抽出回路48に送る。

ウィンドウサイズ抽出回路48は、受信ソケットバッファ22の空き容量を示したウィンドウサイズを送信パケットのトランスポート層ヘッダから抽出し、受信バッファ長計算回路49に渡す（同S32）。

【0091】

受信バッファ長計算回路49は、プロトコル処理部21の受信ソケットバッファ22がACKパケットを送信する条件を満たすデータ量を計算し（同S33）、受信バッファ長書込回路50は、計算結果をバッファリング閾値93として、サービスリスト43に書き込む。

【0092】

その後、送信パケット94は、送信インタフェース46を経由し物理ネットワークに送出される。

受信インタフェース31が、物理ネットワークから受信したパケット90の処理動作は、図5で示したステップS10～S27と同じである。

【0093】

但し、ステップS25で参照するサービスリスト43のバッファリング閾値93は、受信バッファ長計算回路49で送信パケット94のウィンドウサイズ情報に基づいて計算されたものである。

この結果、受信ソケットバッファ22の空き容量に応じてビッグパケットのサイズをダイナミックに変更することが可能となると共に、受信ソケットバッファ22に送られたビッグパケットは、直ちに、アプリケーションバッファ12に送られることになり、OSカーネル20におけるパケット処理時間が減少されることになる

【0094】

図8は、本発明のパケット処理装置10の実施例(3)を示しており、このパケット処理装置10が、実施例(1)のパケット処理装置10と異なる点は、受信ソケットバッファが無いこと、再構成バッファ36のビッグパケット内データを、DMA転送回路41を経由して、トランスポート層のフロー制御を行わずにアプリケーションバッファ12に送信すること、及びアプリケーションバッファ12の空き容量93をサービスリスト43に送ることである。

【0095】

図9は、実施例(3)の動作手順を示したフローチャート図である。このフローチャートを参照して実施例(3)の動作を以下に説明する。

本実施例(3)のトランスポート層プロトコル処理部21は、コネクション毎のフロー制御を行っていないが、ポート番号等により複数の宛先アプリケーションを識別できる。プロトコル処理部11は、このプロトコル処理部21を採用するOSカーネル20上で通信を行う。

【0096】

また、本実施例(3)の説明は、他のホスト上のアプリケーションから、本アプリケーションプロトコル処理部11にデータが送信されている場合について述べる。

こうしたトランスポート層プロトコルではコネクションレス通信（例えば、UDP）と考えられるので、送信元と宛先のプロセス間で行われる通信はセッションと呼ばれる。

【0097】

まず、プロトコル処理部11は、通信を行う前にパケット（ビッグパケット）内データを受信するアプリケーションバッファ12を準備する。そして、アプリケーションバッファ12の容量により、再構成バッファ36のバッファリング閾値93を決定し、システムコールによりOSカーネル20に通知する。

【0098】

OSカーネル20は、ドライバ25を経由してインタフェースモジュール30のハー

ドウェアメモリ上にあるサービスリスト43にバッファリング閾値93を書き込む。
この書込は、初回だけでもよいし、頻繁に行ってもよい。

プロトコル処理部21は、ドライバに依頼し、サービスリスト43に通信セッションの識別情報を書き込む（同図のステップS40）。

【0099】

受信インタフェース31は、パケット物理ネットワークから到着する受信パケット90を待ち、受信したパケット90をコネクション識別回路32に渡す（同S41及びS42）。なお、識別回路32は、セッション識別回路と称すべきであるが、実施例(1)及び(2)のコネクション識別回路32と、使用目的と動作内容が同じであるので、便宜上、コネクション識別回路32と称することにする。これはコネクションキューについても同様である。

【0100】

以後のステップS43～S59の動作は、図5に示したステップS11～S27と同じである。

但し、ステップS44において、コネクション識別回路32は、受信パケット90が、管理しているセッションのパケットであるか否かを、パケット90のネットワーク層及びデータリン層のヘッダのセッション識別情報をキーとしてサービスリスト43を検索して識別する。

【0101】

また、ステップS57で用いるバッファリング閾値93は、アプリケーションバッファ12の空き容量93である。

また、ステップS59では、OSに割り込みを上げることにより、再構成バッファ36のビッグパケットは、トランスポート層ヘッダが外され、パケット内データはアプリケーションバッファ12に転送される。

【0102】

また、インタフェースモジュール30は、遅延を許さない非蓄積パケットが到着した場合は、蓄積せずにOSに送信し、再構成バッファに蓄積中のビッグパケットもOSに即時に送信することは実施例(1)と同じである。

この結果、実施例(1)と比較して、さらに高速で、アプリケーションバッファ

にパケットが伝送されることになる。

【0103】

図10は、本発明のパケット処理装置10の実施例(4)を示している。このパケット処理装置10の構成は、図6に示した実施例(2)のパケット処理装置10のインタフェースモジュール30内に、図20のL3スイッチ81を設けた構成になっている。

この構成にするために、コネクション識別回路32は、ルーティング処理部75からの自局宛受信パケットを入力し、送信バッファ45からの送信パケットは、出力側SW部79に出力されている。

【0104】

また、パケット処理装置10の再構成バッファ36及び送信バッファ45は、L3スイッチ81の共有メモリ78に配置され、受信キュー40及び送信キュー44は、バッファ管理回路37に配置されている。

なお、図10においては、実施例(2)の上位層のOSカーネル20及びアプリケーション層プロトコル処理部11は、モジュール30とバス73で接続されたCPU71及びメインメモリ72として図示されている。

【0105】

図11は、実施例(4)におけるL3スイッチのメモリマップ例を示しており、レジスタ51、サービスリスト43、コネクションキュー42、受信キュー40、及び再構成バッファ36は、それぞれ、0x000000~0x00ffff, 0x010000~0x01ffff, 0x020000~0x02ffff, 0x030000~0x033fff, 及び0x034000~0x133fffに割り付けられてる。なお、同図には、送信バッファ45及び送信キュー44は、図を省略している。

【0106】

図12は、再構成バッファの構成例を示しており、この再構成バッファ(空間)36には、次に読出を開始するアドレスを指し示すリードポイントと、次に書込を開始するアドレスを指し示すライトポイントが設定されている。

また、再構成バッファ36で再構成されるビッグパケット92は、ヘッダ92_1及びデータ92_2で構成されている。ヘッダ92_1は、TCPヘッダ及びIPヘッダを組み合わせたものである。

【0107】

図13は、受信キュー40のフォーマット例を示しており、この受信キュー40は、ポインタアドレス、ホストメモリアドレス、及びビッグパケット長で構成されている。なお、ビット数の説明は省略する（以下、コネクションキュー42、サービスリスト43、ビッグパケット92の説明についても同じ）。

【0108】

図14は、コネクションキュー42のフォーマット例を示しており、このコネクションキュー42は、送信元IPアドレス、宛先IPアドレス、送信元ポート番号、宛先ポート番号、ビッグパケット開始アドレス、パケット全長、及びパケット番号で構成されている。

【0109】

図15は、サービスリスト43のエントリのフォーマット例を示しており、同図(1)に示すように、各エントリは、コネクション情報である送信元IPアドレス、宛先IPアドレス、送信元ポート番号、宛先ポート番号、プロトコル、バッファサイズ（受信ソケットバッファの最大値）、及びコネクションキューアドレスから成っている。このエントリは、コネクションに対応しており、サービスリスト43には、コネクションが作成された順に書き込まれる。

【0110】

サービスリスト43のプロトコルフィールドには、IPヘッダに書き込まれる上位層のプロトコルヘッダ種類を指定する番号を書き込む。

同図(2)は、上位層のプロトコルとその番号を示しており、本実施例(4)では、TCPプロトコルを用いるので、tcp=“6”がプロトコルフィールドに書き込まれる。

【0111】

図16(1)は、ビッグパケット92のフォーマット例を示している。このビッグパケット92は、IPヘッダ92_3及びTCPヘッダ92_4から成るビッグパケットヘッダ92_1並びにデータ92_2で構成されている。

IPヘッダ92_3及びTCPヘッダ92_4のフォーマットは、それぞれ標準のIPヘッダ3及びTCPヘッダと同一であるので説明を省略する。

【0112】

同図(2)は、TCPヘッダ92_4のコントロールフラグ部の構成を示しており、このフラグ部は、URG、ACK、PSH、RST、SYN、及びFINフラグで構成されている。

同図(3)は、後述するチェックサム計算の際に用いられる疑似ヘッダを示している。この疑似ヘッダは、送信元アドレス、宛先アドレス、プロトコル・タイプ、TCP長、及び“0x00”が書き込まれるフィールドで構成されている。

【0113】

以下に、本実施例(4)の動作手順を説明する。なお、本実施例の動作手順は、実施例(1)及び実施例(2)を組み合わせたものであるので、フローチャート図は省略する。

また、本実施例(4)では、TCP及びIPの両プロトコルを用いて通信を行っているファイアウォールを例にとり、物理的なネットワークとしてイーサネットを想定して説明する。

【0114】

まず、送信元ホストとファイアウォールとの間にコネクションが確立される。

サーバの上位層のOSカーネル20(図6参照)は、コネクション毎にTCPパケットを管理する受信バッファ22を割り当てる。受信バッファは受信ソケットバッファと呼ばれ、その最大値はOSカーネル20の設定値により決まっている。また、送信元ホストのOSカーネルにおいてもコネクションを確立すると共に、送信ソケットバッファが作成され、その最大値はOSカーネルの設定値により決まっている。

【0115】

OSカーネル20は、新たに確立されたコネクションの情報である送信元IPアドレス、送信元ポート番号、宛先IPアドレス、宛先ポート番号、プロトコル番号、及び受信ソケットバッファの初期値である受信バッファサイズ(空き容量)93をインタフェースモジュール30に書き込むようにドライバ25(図6参照)に依頼する。

【0116】

ドライバ25は、ハードウェアレジスタ51(図11参照)の一つ(servtop)を見て、情報を書き込むサービスリスト43のアドレスを知り、コネクション情報96及び

受信バッファサイズ93をインタフェースモジュール30のサービスリスト43（図15(1)参照）に書き込む。

【0117】

TCPプロトコルではデータパケットを受信すると、受信できたデータに対し、シーケンス番号、及び現在の受信バッファの空き容量をウィンドウとして含むACKパケットをインタフェースモジュール30に送信する。

インタフェースモジュール30は、ACKパケット内の情報を収集し、上位層のバッファの空き状態を知り、以降の送信元ホストからの受信パケットをビッグパケット化する際にそのサイズを動的に変更する。

【0118】

すなわち、インタフェースモジュール30において、OSカーネル20から送信されたACKパケットは、DMA転送回路41及び送信バッファ45を経由してコネクション識別回路47に送られる。

コネクション識別回路47は、サービスリスト43のコネクション情報とパケットのコネクション情報（送信元IPアドレス、宛先IPアドレス、送信元ポート番号、及び宛先ポート番号等）とが一致したことを検出した場合、パケットが自局から送出するTCPパケットであることを確認する。この後、ACKパケットであることをチェックするために、IPヘッダの先頭から10ワード目（図16(1)及び(2)参照）を読み込み、11ビット目のACKフラグ＝“1”であるか否かを判定する。ACKフラグ＝“1”であるとき、図7に示した動作手順と同じ手順が実行される。

【0119】

すなわち、ウィンドウサイズ抽出回路48が、16～31ビットのウィンドウフィールドからウィンドウサイズを抽出、受信バッファ長計算回路49がウィンドウサイズから受信バッファ長（空き容量）93を計算し、受信バッファ長書込回路50がサービスリスト43のバッファサイズフィールドに計算結果を書き込む。

【0120】

一方、送信元ホストからインタフェースモジュール30が、受信インタフェース31を経由して受信パケット90を受信すると、ルーティング処理部75は、自局宛のIPパケットか否かを判定し、自局宛のパケットをコネクション識別回路32に送る

【0121】

コネクション識別回路32は、TCP/IPヘッダの3ワード目9～15ビットのプロトコル番号、4，5ワード目の送信元IPアドレス、宛先IPアドレス及び7ワード目の送信元ポート番号、宛先ポート番号を、サービスリスト43の対応データと比較して、管理しているコネクションのパケットであるか否かを識別する。

【0122】

管理すべきパケットであるとき、10ワード目の10～15ビットのURG，ACK，PSH，RST，SYN，及びFINフラグの中の1つでも“1”が立っている場合、即時に、上位層に届くことが期待されるパケットであるため、再構成バッファ36で蓄積せず、受信キュー40に直接入れる。

【0123】

フラグが“1”以外のパケットは、図5で示した動作手順のステップS13～S27が実行され、該パケットは、コネクション毎の再構成バッファ36に送られビッグパケット化された後、上位層の受信バッファに書き込まれる。

すなわち、チェックサム計算回路33は、図16(3)に示した疑似ヘッダを作成し、疑似ヘッダ、TCPヘッダ、及びデータ部を読み込んでTCPチェックサムを計算し、TCPヘッダ5ワード目の0～15ビットのチェックサムと比較する。不一致の場合は正常なパケットでないとして廃棄し、それまでに再構成バッファ36に蓄積したビッグパケットを受信キュー40にキューイングする。

【0124】

一致した場合は正常なパケットであるとして、再構成バッファ書込回路35は、データを再構成バッファ36に書き込むと同時に、バッファサイズ及びパケット数をコネクションキュー42のパケット全長フィールド、パケット番号フィールドにそれぞれに書き込む。

【0125】

そして、バッファ管理回路37にパケットを書き込むアドレスを指定するように依頼すると共に、バッファ条件判定回路38に書き込んだことを通知する。

バッファ条件判定回路38は、全長フィールドをサービスリスト43のバッファサ

イズフィールドと比較し、バッファサイズ以上になった場合、受信キュー40にビッグ packets をキューイングし、OSカーネル20に割込をかける。

【0126】

図12に示す如く、再構成バッファ36は、コネクション毎にビッグ packets 92をバッファリングする。再構成バッファ36は、バッファ管理回路37によりリング状にオーバーラップ可能なように構成されており、読出済みアドレスを示すリードポインタと書込済みアドレスを示すライトポインタにより管理される。

【0127】

バッファ管理回路37は、バッファ書込回路35からの依頼により入力すると書込開始アドレスを出力する。バッファ管理回路37では一つのコネクションの再構成バッファとして、例えば64Kバイトのバッファ領域を確保し、ライトポインタを修正する。書込済みポインタがオーバーラップしてリードポインタを越えてしまった場合はエラーとして“0x0”のアドレスを出力する。

【0128】

再構成バッファ書込回路35は、バッファ管理回路37からアドレスが取得できた場合は、コネクションキュー42（図14参照）のビッグ packets 開始アドレスにビッグ packets の開始位置を記入し、 packets を指定された再構成バッファ36の開始アドレスから書き込む。

【0129】

なお、ヘッダ削除回路34は、コネクションキュー42の packets 番号を見て、“0”の場合は最初の packets であると判断して、TCP/IPヘッダも含めて再構成バッファ書込回路35に渡し、 packets 番号が“1”以上の場合は二つ目以降の packets であると判断し、ヘッダ部を削除してデータ部分のみ再構成バッファ書込回路35に渡す。

【0130】

また、再構成バッファ書込回路35は、再構成バッファ36に最初の packets の書き込みを行った場合は、タイマ39を一定時間後にタイムアウトするようにセットする。そして、 packets を再構成バッファ36に書き込む毎に、コネクションキュー42内の packets 番号を“1”だけカウントアップする。

【0131】

二つ目以降の各パケットを再構成バッファ36に書き込み時には、削除するIPパケットのヘッダに記述されているパケット全長及びヘッダ長から計算した“データ長”、及びTCPヘッダ中の“シーケンス番号”を読み込む。

そして、最初のパケットと共に再構成バッファに書き込まれたIPヘッダ中の「パケット全長」に“データ長”を加算し、TCPヘッダ中の「シーケンス番号」を、上記の“シーケンス番号”に書き換える。さらにコネクションキュー42のパケット全長のエントリを加算結果の「パケット全長」に書き換える。

【0132】

その後、バッファ条件判定回路38は、コネクションキュー42の“パケット全長”とサービスリスト43内の“バッファサイズ（空き容量）”を比較し、“パケット全長”が“バッファサイズ”を越えている場合、ビッグパケットを受信キュー40にキューイングし、CPU71に割込を掛ける。

【0133】

受信キュー40（図13参照）には、再構成バッファ36内のパケット開始位置（コネクションキュー42のビッグパケット開始アドレス）を記入する。

以後、ビッグパケットは、受信キュー40にキューイングされた順番に従って、DMA転送回路41を介して受信ソケットバッファに送信される。

【0134】

二つ目以降のデータパケットがなかなか到着せず、ビッグパケットのサイズがバッファサイズを越えない場合、一定時間が経過するとタイマ39がタイムアウトし、バッファ条件判定回路38に通知する。バッファ条件判定回路38は、バッファサイズに達していなくても、再構成バッファ36内のデータ開始アドレスを受信キュー40に書き込み、コネクションキュー42の対応するコネクションのキューをデキューする。OSに対しては、ビッグパケットを受信したことを知らせる割込を発行する。

【0135】

OSカーネル20のドライバ25は、インタフェースモジュール30より割込を受けて、管理バッファ（ここではBSD系のUNIX OSに見られるmbuf等を想定している）

をOSカーネル20が管理するメインメモリに獲得し、データをコピーすべきアドレスを得て（図6参照）、受信キュー40内のホストメモリアドレスに書き込む。

【0136】

DMA転送回路41はDMA転送を開始し、受信キュー40に示された、ポインタアドレスからホストメモリアドレスにビッグパケット長のデータを転送する。

OSカーネル20は、受信したビッグパケットを含むmbufを一つのパケットとして通常通りプロトコル処理する。TCPソケットバッファで一旦受信するが、上述したように、受信したビッグパケットのデータ量は受信ソケットバッファのACKパケット送出条件を満たしているので、即時に、ACKパケットが作成され、インタフェースモジュール30を経由して送信元ホストへ送信されると共に、ビッグパケットはアプリケーションバッファに転送される。

【0137】

上述したように、図15に示したサービスリスト43のバッファサイズは、OSカーネル20のTCPプロトコルから返送されるACKパケット内のウィンドウフィールドのデータに基づき動的に変更される。

以下に、ウィンドウフィールドのデータに基づいてバッファサイズを計算する手順を図17を参照して説明する。

【0138】

図17(1)に示すように、OSカーネル20におけるTCPプロトコルは、受信ソケットバッファを二つのrcv_adv及びrcv_nextポインタで管理している。

rcv_advポインタは、受信側ソケットバッファで受信可能であることを送信側にACKパケットを用いて既に通知済であるシーケンス番号であり、rcv_nextポインタは送信側TCPから受信したデータのシーケンス番号に1を足した値、つまり次のデータを受信開始する位置を示している。

【0139】

TCPプロトコルでは、パケット受信毎にウィンドウwin=(最大ソケットバッファ長)-{(rcv_advポインタが指し示すアドレス)-(rcv_nextポインタが指し示すアドレス)}を計算し、ウィンドウwinが、最大バッファ長の50%以上、或はウィンドウwinが最大セグメント長の2個分の条件を満たすとウィンドウwinをウィン

ドウサイズとしたACKパケットを送信する。

【0140】

この条件によれば、少なくとも最大バッファ長の1/2のデータを受信すれば必ず、ACKパケットを送出する。

インタフェースモジュール30で再構成されビッグパケット化したパケットをOSカーネル20に渡した場合に必ずACKパケットをが発行されるようにACKパケット送出条件を満たすようなサイズにパケットをビッグパケット化する。

【0141】

これは、OS内の受信ソケットバッファで再度蓄積されずにアプリケーションバッファに直ちにコピーするためである。

ここでは以下の方法でこれを実現する。同図(2)に示すように、ウィンドウwinが最大バッファサイズの半分より小さい場合は、(最大バッファサイズ)-(ウィンドウwin)のデータを受信すれば少なくともACKパケットを返信する。また、同図(3)に示すようにウィンドウwinが最大バッファ長の半分よりも大きい場合は、最大バッファ長の半分のデータを受信すればACKパケットを返信する。

【0142】

そこで、次式(1)により、再構成バッファ内に蓄積しビッグパケット化するデータ量(M)を決定する。

データ量M

$$= \text{MAX}[(\text{最大バッファ長}-\text{ウィンドウwin}), (\text{最大バッファ長}/2)]$$

・・・式(1)

このデータ量Mを、受信バッファ長計算回路49が計算し、受信バッファ長書込回路50がサービスリスト43のバッファサイズフィールドに書き込む。

【0143】

また、TCPプロトコルでは一般に、ソケットバッファにデータパケットを受信し、一定時間確認応答の送信をペンディングしている時(delayed ACKパケット)に、逆方向の(送信元ホストへの)データパケットを送信することになった場合、このデータパケットに確認応答のフラグも立てて送信することで、delayed ACKタイマのタイムアウトを待たずに確認応答を行う。このように逆方向のデータが

流れている場合は、確認応答を送信できる状態であるので、いつまでも再構成バッファ36に蓄積しておくのは好ましくない。

【0144】

そこで、コネクション識別回路47により逆方向へのデータパケットが送信キュー44にキューイングされたことを認識した場合は、受信バッファ長書込回路50は、タイマ39へ通知し、そのコネクションに関するタイマを強制的にタイムアウトさせる。

【0145】

これにより、バッファ条件判定回路38は、その時点までに再構成バッファ36に蓄積したデータパケットをビッグパケット化した後、再構成バッファ36内のデータ開始アドレス(コネクションキューに記入してある)を受信キュー40に書き込み、コネクションキュー42の対応するコネクションのキューをデキューする。また、OSカーネル20に対し、ビッグパケットを受信したことを知らせる割込を発行する。

【0146】

また、送信元ホストから図16(2)のRST及びFINフラグがそれぞれ立っているFINパケット及びRSTパケットのいずれかを受信してコネクションが切断された場合、OSカーネル20は、ドライバ25を介して、サービスリスト43にアクセスし、該当するコネクションを削除する。

【0147】

必ずしもリストの最後から削除するわけではないので、該当エントリ以降のエントリを一つづつ上に詰める処理を行った後、新たなサービスリスト43の書き込み位置を示すハードウェアレジスタ(servtop)を更新する。

この結果、OSカーネル20に渡すパケット数、従って割込数が減り、OSカーネル20内プロトコル処理の削減、割込処理のオーバヘッドの削減によるOS内処理の高速化が可能となる。

【0148】

また、OSカーネル20からのインタフェースモジュール30のハードウェア資源にアクセスすることにより、ハードウェアL3フォワーディング処理の妨害を回避

することが可能となる。

また、本発明によれば、OSとアプリケーションのインタフェース(API)には手を加えないため、アプリケーションの変更なしにアプリケーション経由のトラフィックを高速に処理することが可能となる。

【0149】

また、本発明はファイアウォール機能だけでなく、L3スイッチ上にプロキシサーバ機能を搭載したことによりプロキシ機能をも高速で処理することが可能となる。

図18は、通信ホスト61、62に、本発明の実施例(5)であるパケット処理装置を含むNIC装置を実装した通信例を示している。

【0150】

すなわち、本発明のパケット処理装置は、通信ホスト61、62における通信インタフェースカード(NIC装置)に適用することも可能である。

同図において、通信ホスト61は、L3スイッチ81、通信網80、及びL3スイッチ82を経由して通信ホスト62に接続されている。

【0151】

各通信ホスト61及び62は、アプリケーション11、OSカーネル20、及びパケット処理部10を含むNIC装置(図示せず)で構成されている。

以下に、通信ホスト61-通信ホスト62間の通信動作を説明する。

(1)通信ホスト61は、TCP/IPプロトコルを用いて、通信ホスト62とコネクションを開設する。

(2)コネクションが開設されると、通信ホスト61はTCP/IPプロトコルに従い、パケットをL3スイッチ81を介して通信網80に送出する。

(3)通信ホスト62は、NIC装置においてパケットを受信すると、実施例(1)と同様な手続きで、受信パケットをNIC装置内に設けられた再構成バッファに蓄積する。

(4)再構成バッファが通信ホスト62におけるTCPプロトコルの動的に変化する受信バッファ量に達すると、再構成バッファ内のパケットを一つのIPパケットに再構成し、割り込みでOSカーネル20に渡す。この再構成の際にチェックサム計算回

路が必要となる。

(5)OSカーネル20においては、受信したパケットに対し、IPプロトコル処理及びTCPプロトコル処理を行う。TCPプロトコル処理は、一般にOSカーネル20内のTCP受信バッファにてパケットの蓄積を行う。

【0152】

TCP受信バッファに、一定の量のパケットが蓄積すると、OSの上位にあるアプリケーションに蓄積したパケットをコピーする。

本発明によれば、NIC装置から受信した再構成後のIPパケットはTCP受信バッファと同じサイズに再構成されているので、この受信バッファに入ると直ちにアプリケーション側にコピーされる。

【0153】

この結果、NIC装置からOSへの割込を減らすことで、OS内の割込処理オーバーヘッドを減らすと共に、NIC装置からOSへコピーされるパケットが、サイズは大きくなるが一つになることでハードウェアのパケット受信をブロッキングする可能性を減らすことができる。

【0154】

また、OS内でのTCP処理においても、本来ならば複数回必要なIPヘッダ処理が1回になることでオーバーヘッドを引き下げることが可能となる。

【0155】

【発明の効果】

以上説明したように、本発明に係るパケット処理装置によれば、空き容量通知部が上位層の該受信バッファの空き容量を通知し、蓄積条件判定部が、ビッグパケットのサイズを該空き容量に基づいて決定し、再構成バッファ処理部が複数の受信パケットを1つの該ビッグパケットに再構成して該受信バッファへ送信するように構成したので、上位層の受信バッファへのパケット到着数を削減することが可能となり、通信プロトコルにおけるパケット送信に必要な処理量を減らすことができる。従って、パケットを高速で伝送することが可能となると共に、割込時に発生するコンテキスト切替等のオーバーヘッドを減らすことが可能となり、上位層におけるパケット処理量を減らすことが可能になる。

【0 1 5 6】

また、ネットワーク層の packets 転送機能をハード化した L 3 スイッチが、受信した自局宛の複数の受信 packets を該再構成バッファ処理部に送信するように構成すれば、L3 スイッチのハードウェア packets 転送処理で使用するバッファメモリに対し、OS カーネルがアクセス(メモリコピー等)することによるブロッキングを少なくすることが可能になり、高速なデータ中継が可能となる。

【0 1 5 7】

さらに、NIC 装置が、packets 処理装置を含み、受信した自局宛の複数の受信 packets を該再構成バッファ処理部に送信するように構成すれば、NIC 装置の上位層の OS は、自局宛の受信 packets の処理回数を減少させることが可能となり、OS のオーバーヘッドを無くすことができる。

【図面の簡単な説明】

【図 1】

本発明に係る packets 処理装置の原理(1)を示したブロック図である。

【図 2】

本発明に係る packets 処理装置の原理(2)を示したブロック図である。

【図 3】

本発明に係る packets 処理装置の原理(3)を示したブロック図である。

【図 4】

本発明に係る packets 処理装置の実施例(1)を示したブロック図である。

【図 5】

実施例(1)の動作手順を示したフローチャート図である。

【図 6】

本発明に係る packets 処理装置の実施例(2)を示したブロック図である。

【図 7】

実施例(2)の動作手順を示したフローチャート図である。

【図 8】

本発明に係る packets 処理装置の実施例(3)を示したブロック図である。

【図 9】

実施例(3)の動作手順を示したフローチャート図である。

【図10】

本発明に係るパケット処理装置の実施例(4)を示したブロック図である。

【図11】

本発明に係るパケット処理装置で用いられる各テーブルを示したメモリマップ図である。

【図12】

本発明に係るパケット処理装置の再構成バッファの構成例を示したフォーマット図である。

【図13】

本発明に係るパケット処理装置における受信キューの構成例を示したフォーマット図である。

【図14】

本発明に係るパケット処理装置におけるコネクションキューの構成例を示したフォーマット図である。

【図15】

本発明に係るパケット処理装置におけるサービスリストの構成例を示したフォーマット図である。

【図16】

本発明に係るパケット処理装置におけるビッグパケットのヘッダ (TCP/IP) の構成例を示したフォーマット図である。

【図17】

TCPプロトコルにおけるACKパケットの送信タイミングを示した図である。

【図18】

本発明に係るパケット処理装置の実施例(5)を示したブロック図である。

【図19】

従来の L 3 スイッチを用いたパケット処理装置を示したブロック図である。

【図20】

従来の L 3 スイッチの構成例を示したブロック図である。

【図21】

従来のL3スイッチを用いた通信ネットワークを示したブロック図である。

【符号の説明】

- 10 パケット処理装置
- 11 アプリケーション層プロトコル処理部
- 12 アプリケーションバッファ
- 20 OSカーネル
- 21 トランスポート層プロトコル処理部
- 22 受信ソケットバッファ、受信バッファ
- 24 ネットワーク層プロトコル処理部
- 25 ドライバ
- 30 インタフェースモジュール
- 31 受信インタフェース
- 32 コネクション識別回路
- 33 チェックサム計算回路
- 34 ヘッダ削除回路
- 35 再構成バッファ書込回路
- 36 再構成バッファ
- 37 バッファ管理回路
- 38 バッファ条件判定回路
- 39 タイマ
- 40 受信キュー
- 41 DMA転送回路
- 42 コネクションキュー
- 43 サービスリスト
- 44 送信キュー
- 45 送信バッファ
- 46 送信インタフェース
- 47 コネクション識別回路

- 48 ウインドウサイズ抽出回路
- 49 受信バッファ長計算回路
- 50 受信バッファ長書込回路
- 51 レジスタ
- 61, 62 ホスト
- 63 サブネット
- 64 プロキシサーバ・ルータ
- 65 インターネット
- 66 プロキシサーバ
- 67 ルータ
- 71 CPU
- 72 メインメモリ
- 73 バス
- 74 CAM
- 75 ルーティング処理部
- 76 ヘッダ書換部
- 77 入力側SW部
- 78 共有メモリ
- 79 出力側SW部
- 80 通信網
- 81, 82 L3スイッチ
- 83 ルーティング処理回路
- 84 SW部
- 85 出力バッファ
- 86 プロトコル処理部
- 90, 91 受信パケット
- 92 ビッグパケット
- 93 空き容量
- 94 逆方向パケット、送信パケット

- 95 送信パケット
- 96 コネクション情報
- 100 上位層プロトコル処理部
- 101 受信バッファ
- 102 受信バッファ空き容量通知部
- 200 中位層プロトコル処理部
- 300 下位層プロトコル処理部
- 301 再構成バッファ処理部
- 302 コネクション識別回路
- 303 蓄積条件判定部
- 304 チェックサム計算回路
- 305 逆方向パケット内情報読出回路

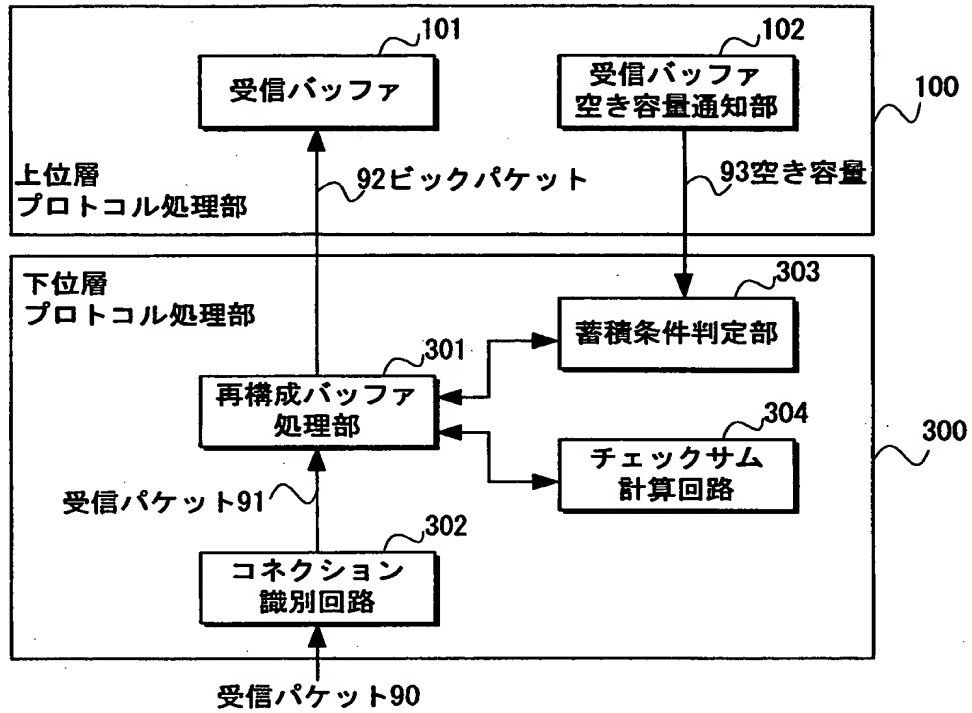
図中、同一符号は同一又は相当部分を示す。

【書類名】 図面

【図 1】

本発明の原理 (1)

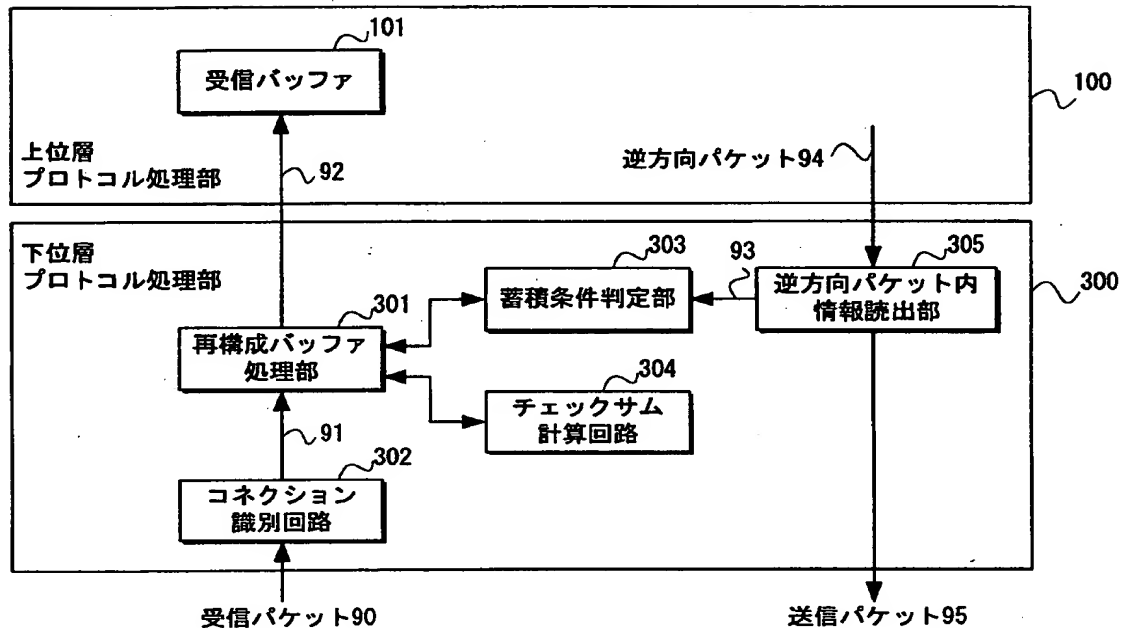
10



【図 2】

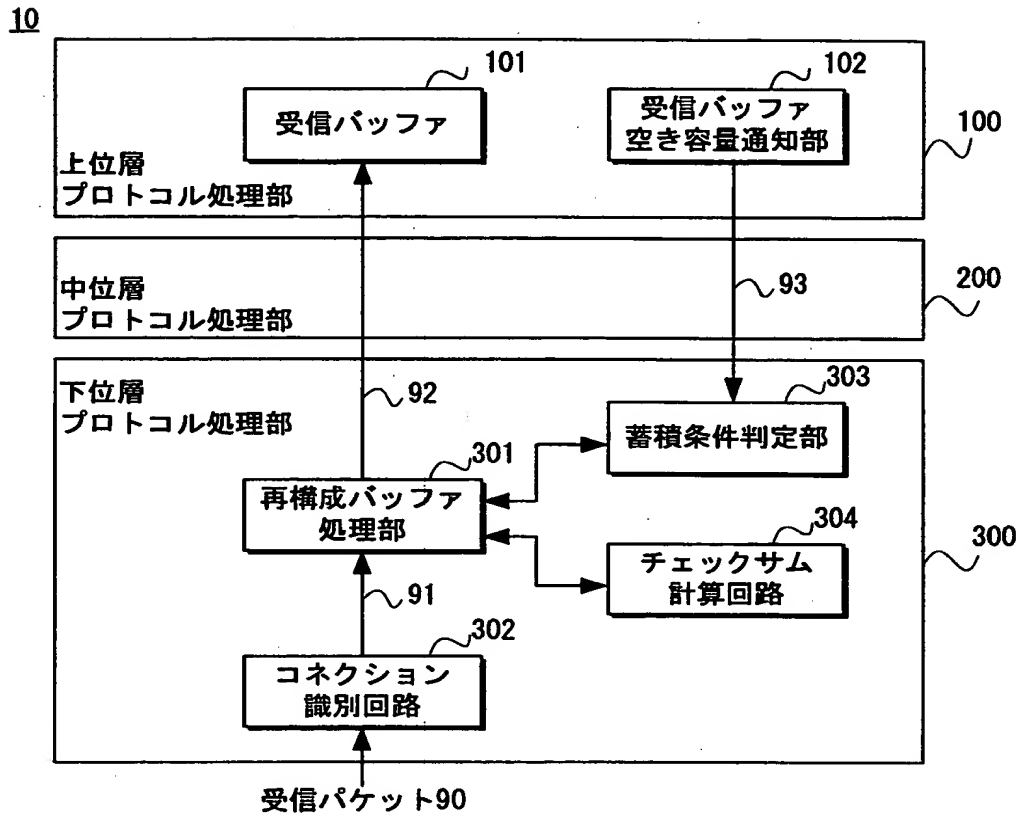
本発明の原理（2）

10



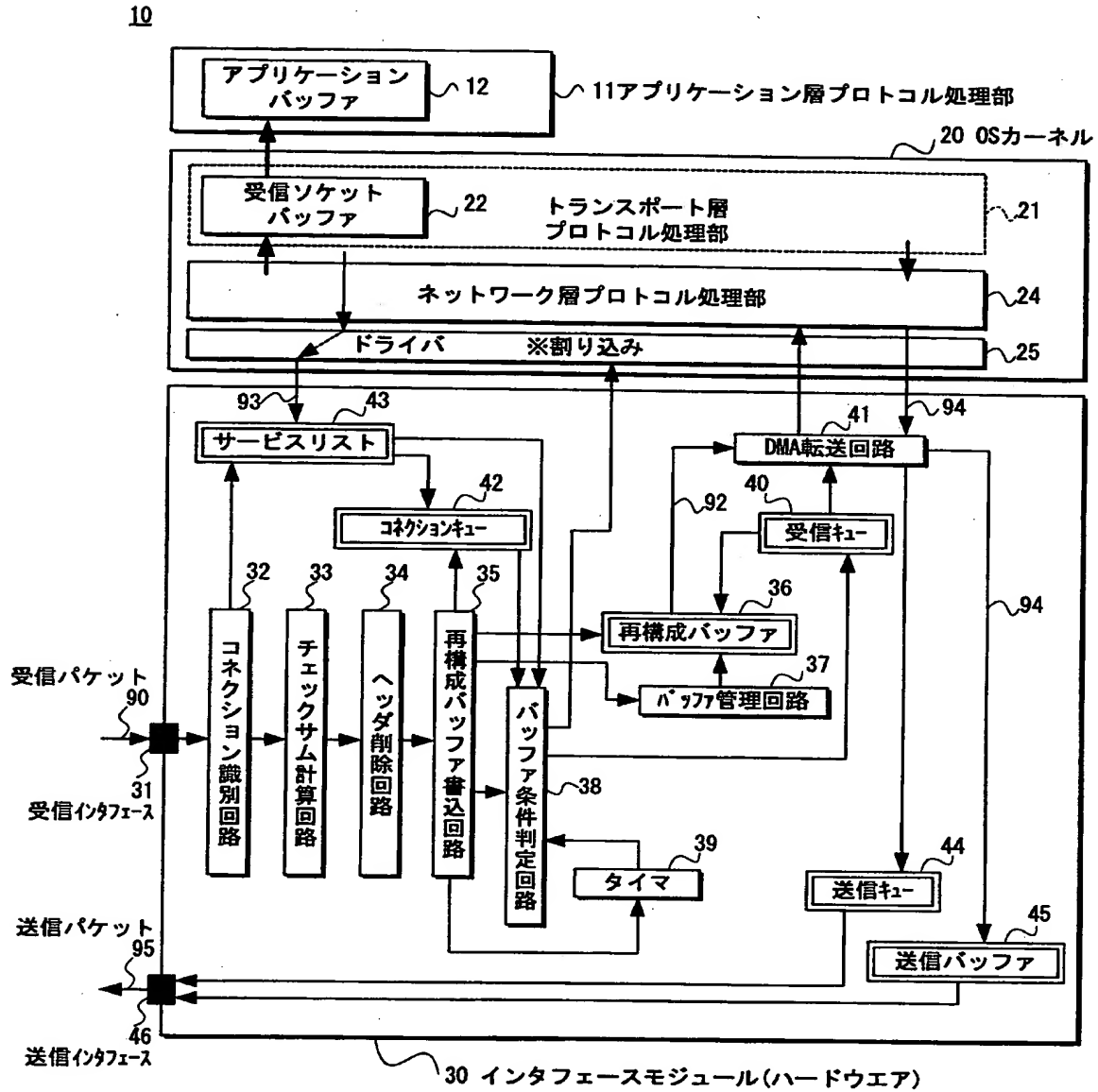
【図 3】

本発明の原理（3）



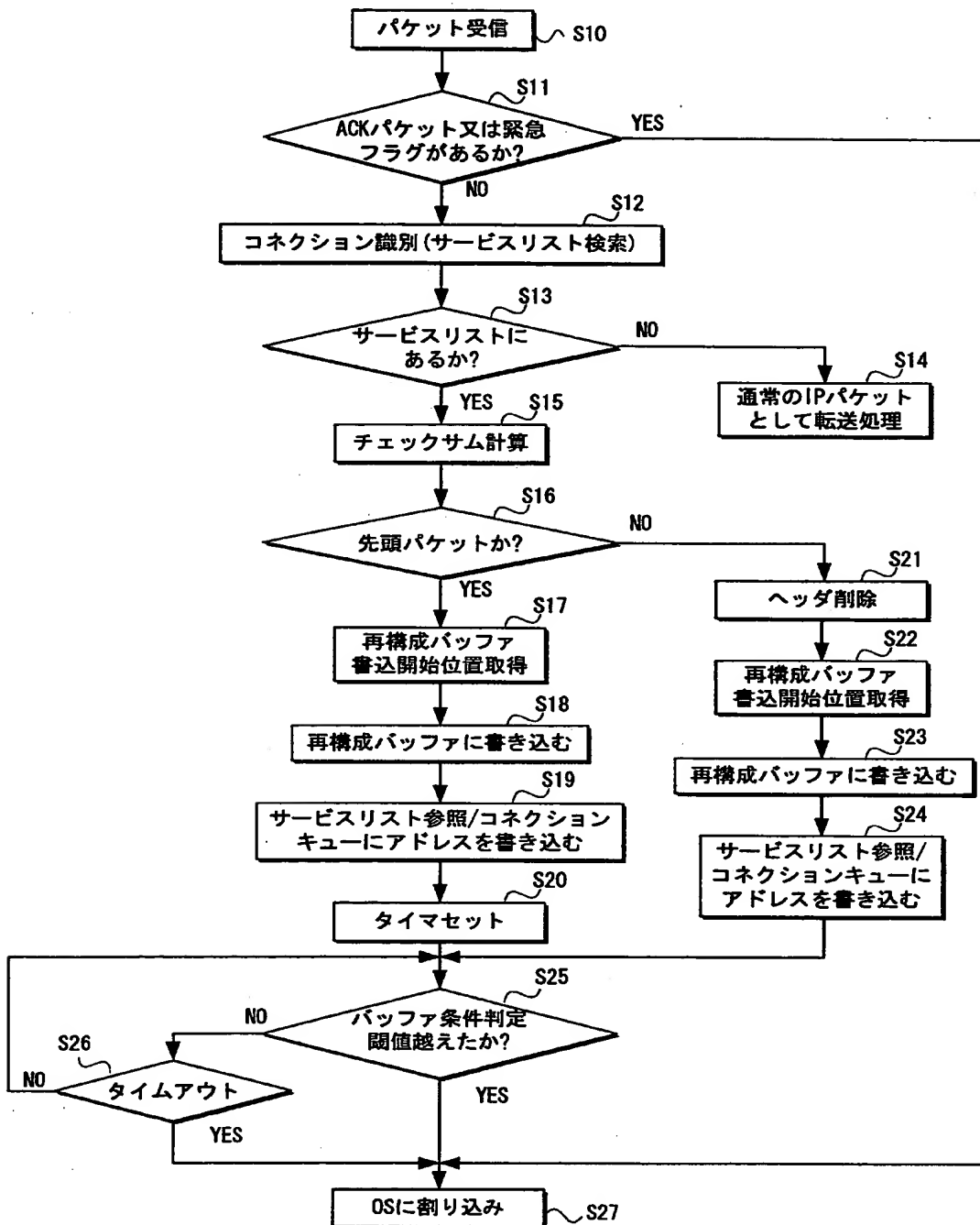
【図 4】

本発明の実施例（１）



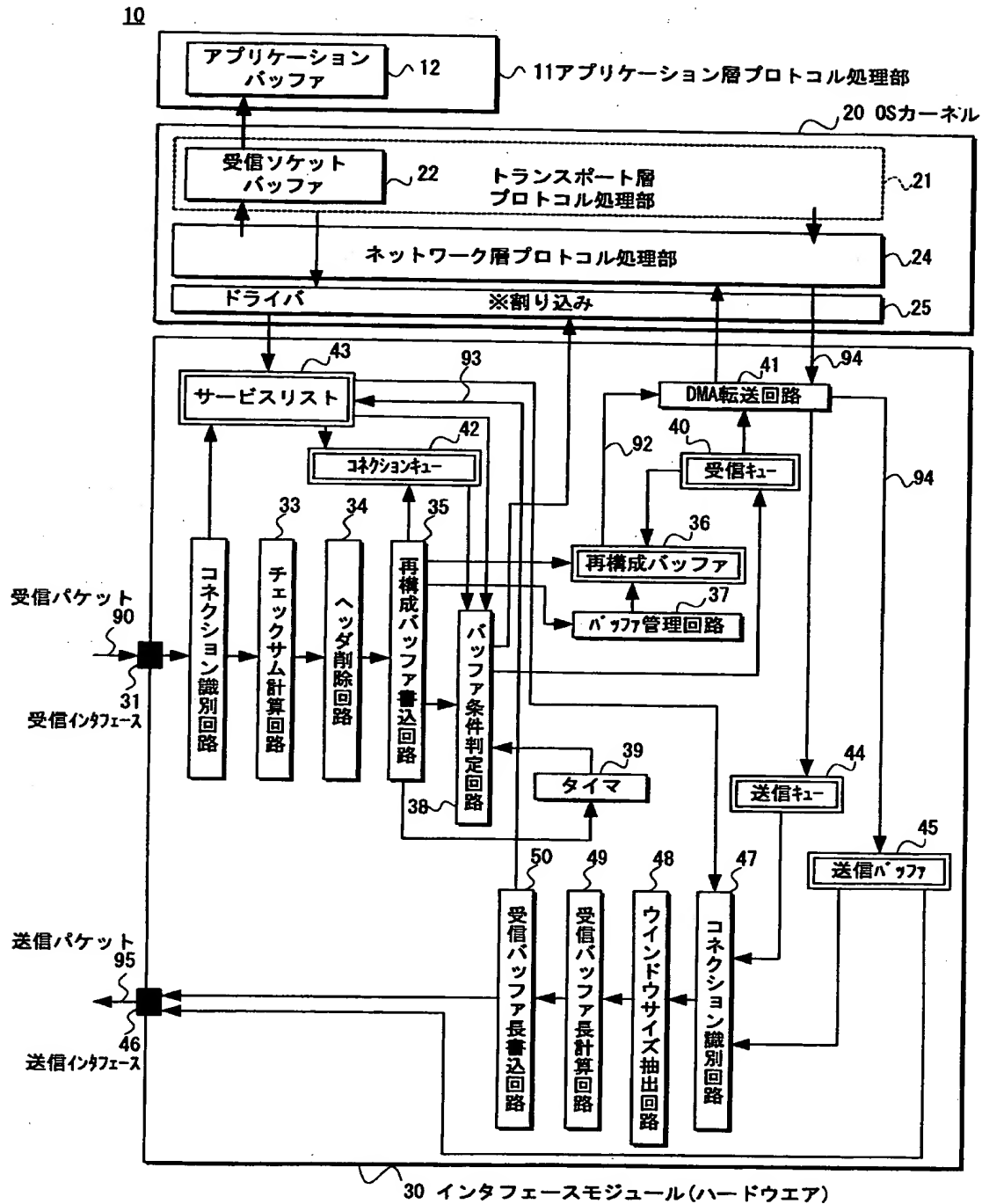
【図 5】

実施例（１）の動作手順



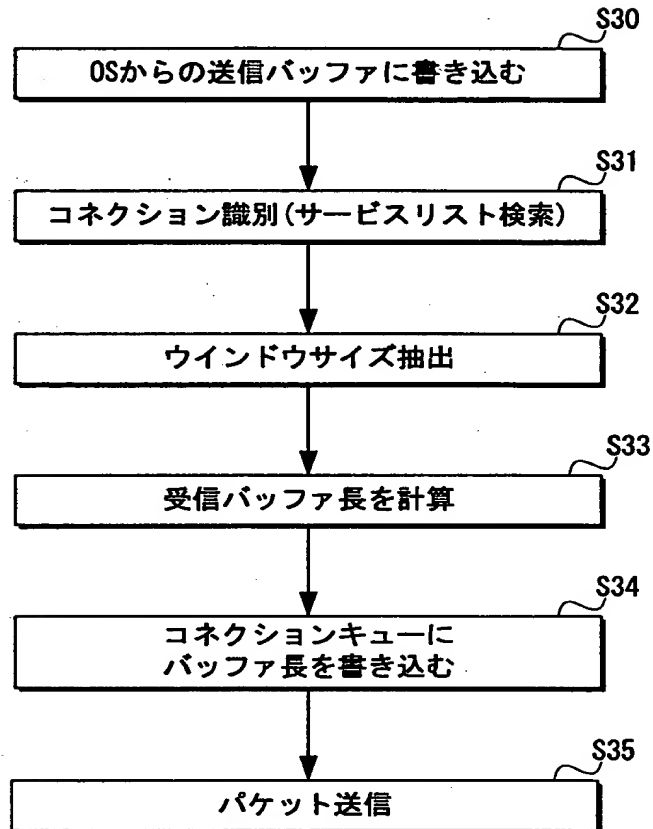
【図 6】

本発明の実施例 (2)



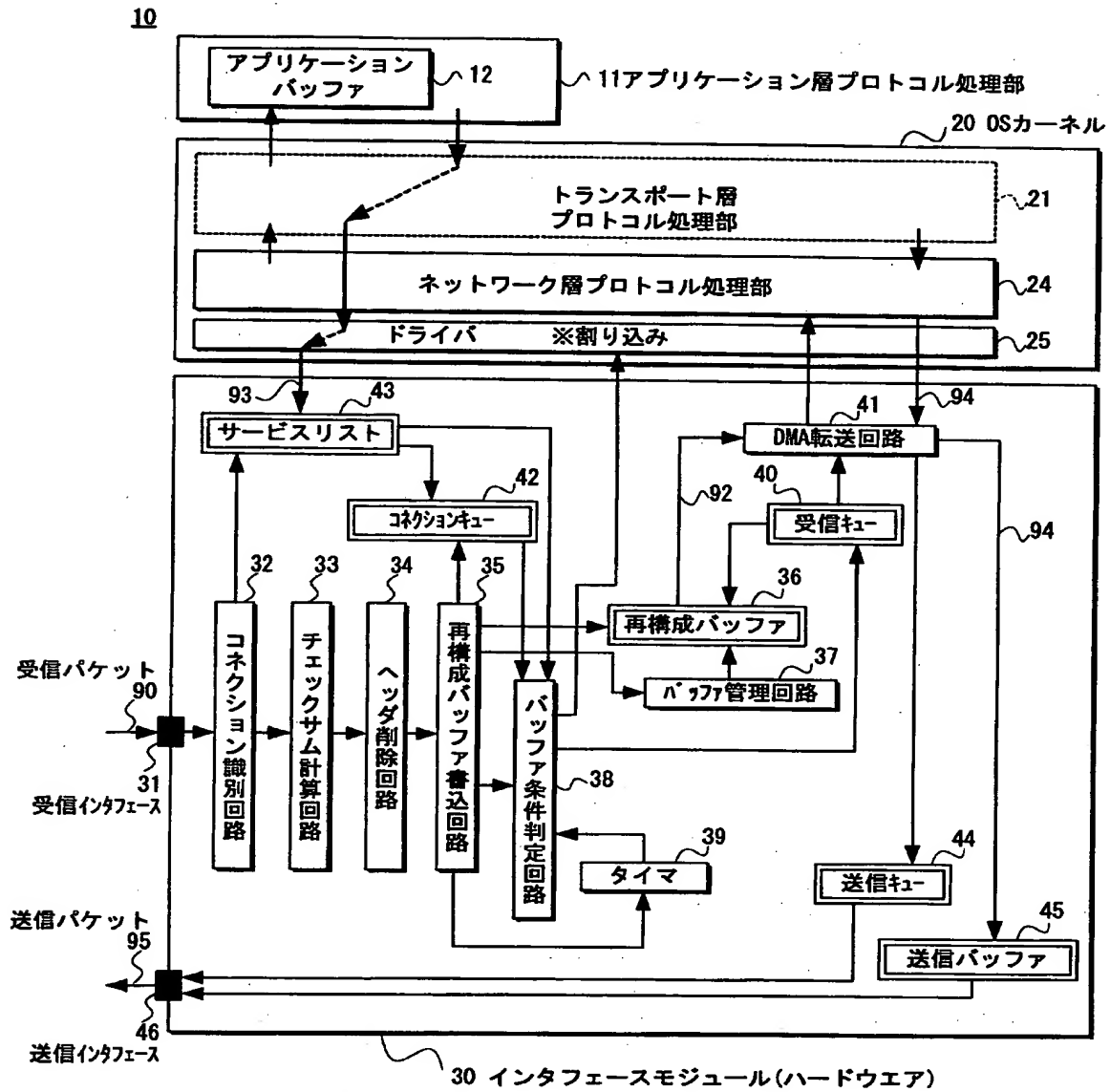
【図 7】

実施例（２）の動作手順



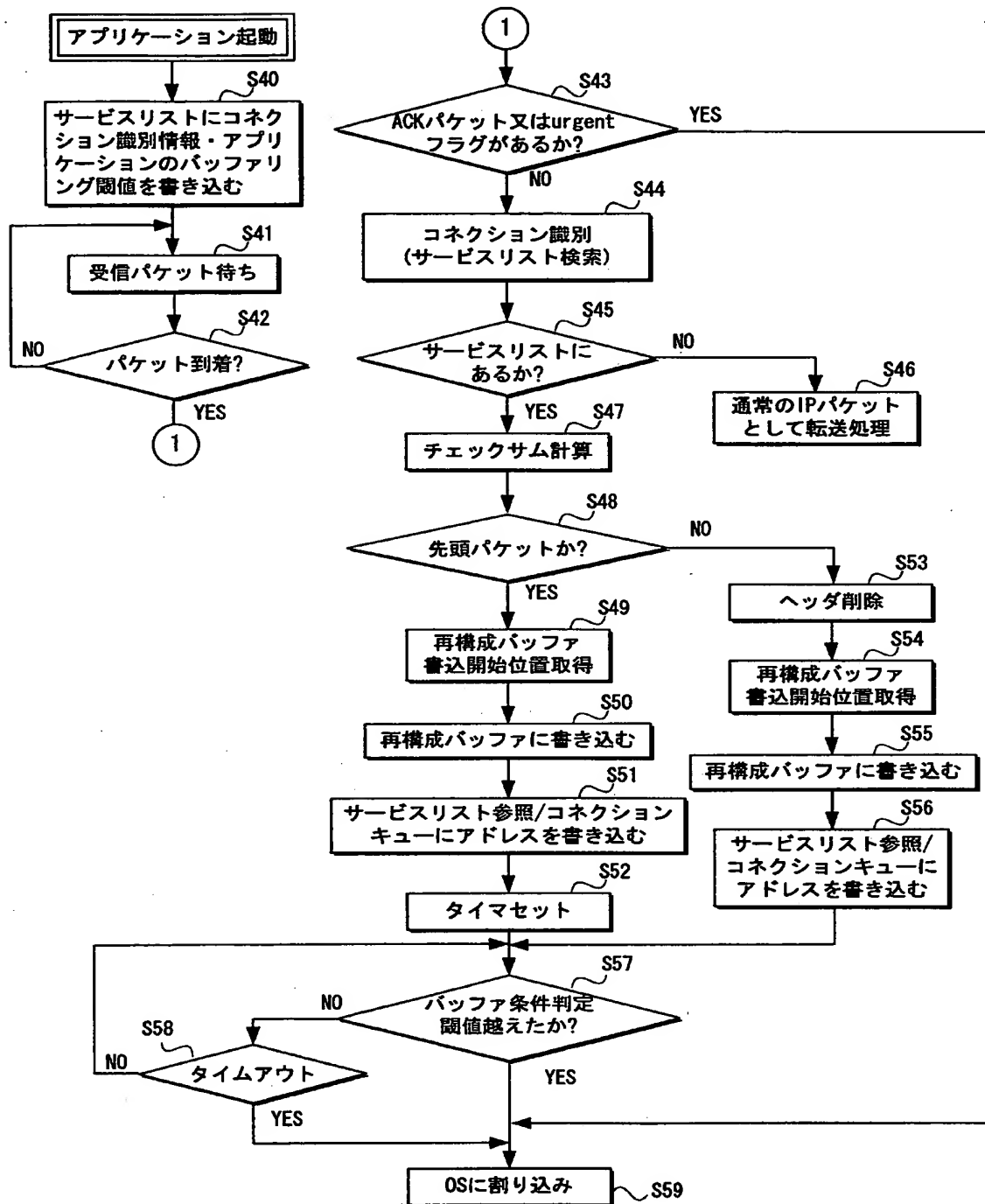
【図 8】

本発明の実施例（3）



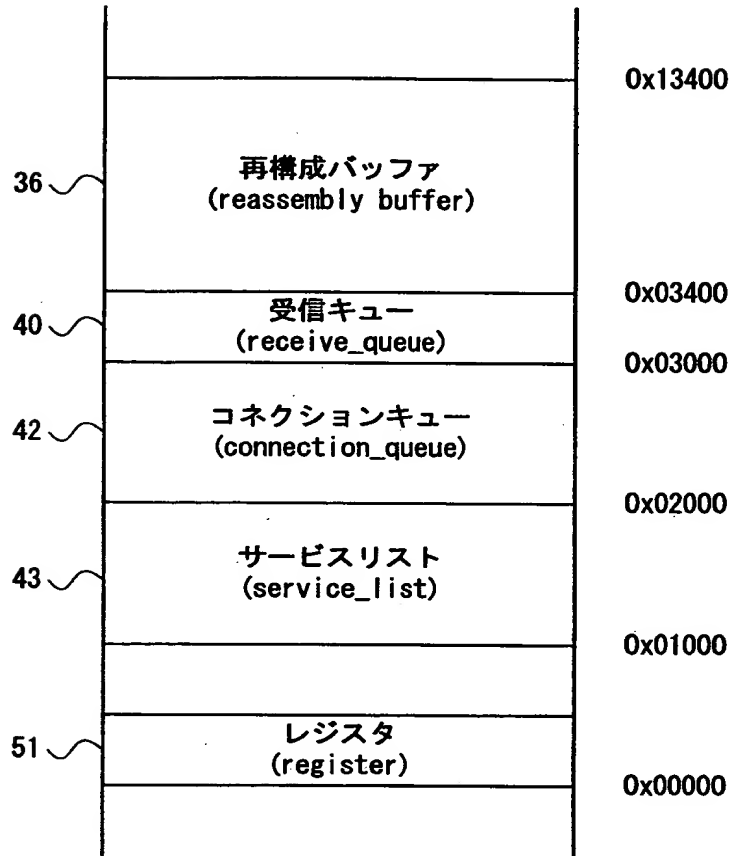
【図 9】

実施例 (3) の動作手順



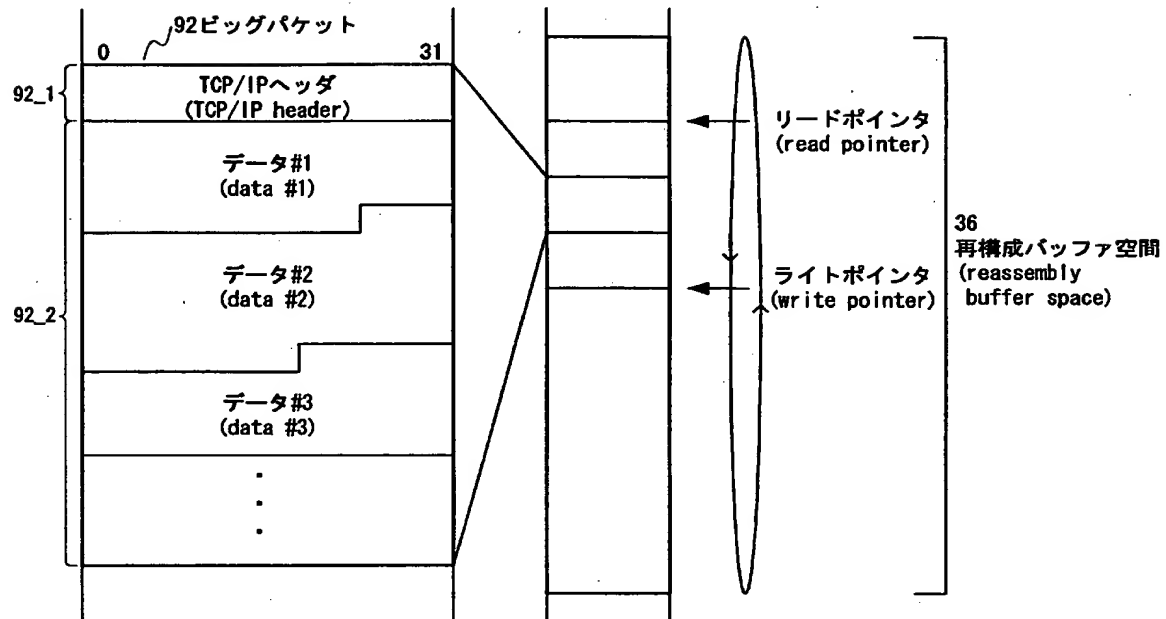
【図 1 1】

L3スイッチのメモリマップ例



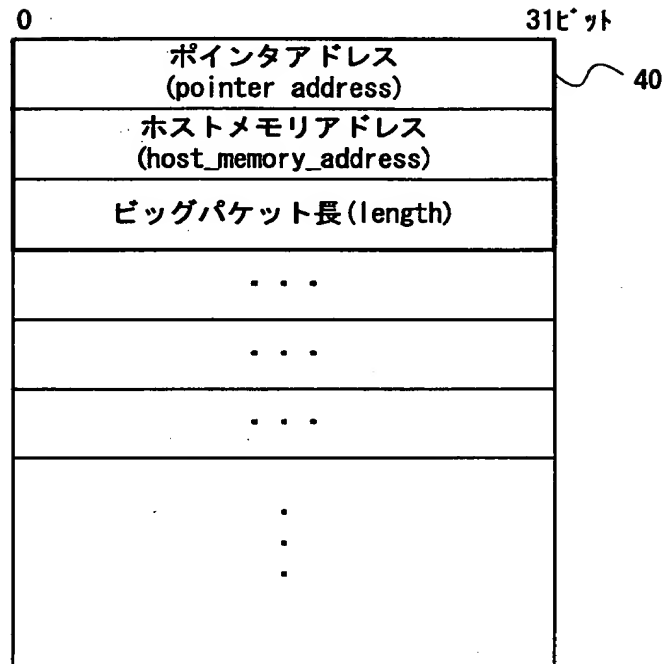
【図 1 2】

再構成バッファの実施例



【図 13】

受信キューのフォーマット例



【図 14】

コネクションキューのフォーマット例

| | |
|---|---------------------------|
| 42 | |
| 送信元IPアドレス (src IP address) | |
| 宛先IPアドレス (dst IP address) | |
| 送信元ポート番号 (src PORT) | 宛先ポート番号 (dst PORT) |
| ビッグパケット開始アドレス (reassemble_start_address) | |
| パケット全長 (total_length) | パケット番号 (packet_number) |

【図 1 5】

サービスリストのフォーマット例

43

(1)

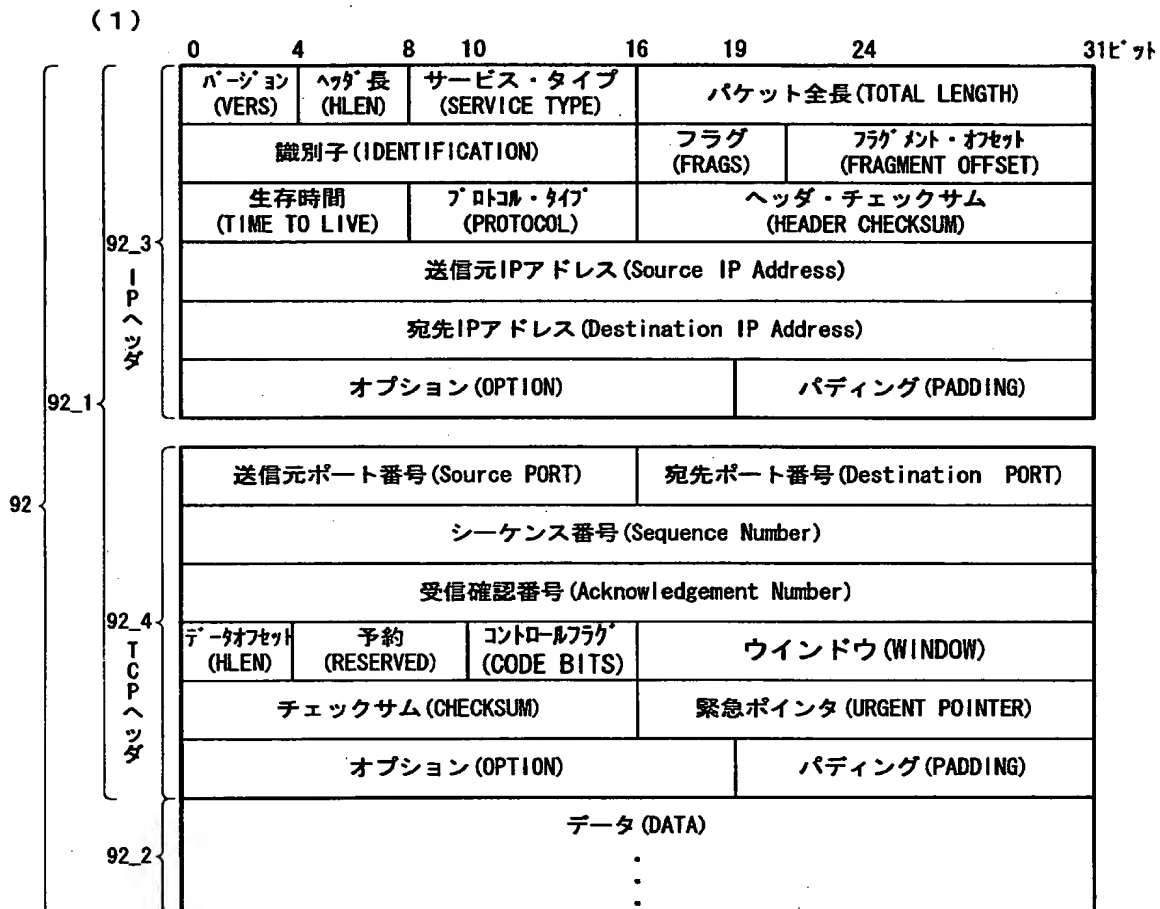
| | | |
|---------------------------|--------------------------|---|
| 送信元IPアドレス(src IP address) | | |
| 宛先IPアドレス(dst IP address) | | |
| 送信元ポート番号(src PORT) | | 宛先ポート番号(dst PORT) |
| プロトコル (protocol) | バッファサイズ (buffer_size) | コネクション キュー アドレス (connection_queue address) |

(2)

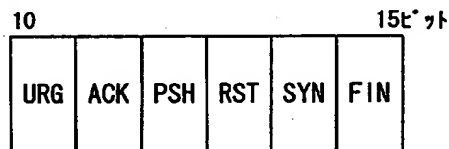
| プロトコル | 番号 | 備 考 |
|-------|----|---|
| icmp | 1 | インターネット・コントロール・メッセージ・プロトコル (internet control message protocol) |
| tcp | 6 | トランスミッション・コントロール・プロトコル (transmission control protocol) |
| egp | 8 | 外部ゲートウェイ・プロトコル (exterior gateway protocol) |
| udp | 17 | ユーザ・データグラム・プロトコル (user datagram protocol) |
| rdp | 27 | リライアブル・データグラム・プロトコル (“reliable datagram” protocol) |

【図 16】

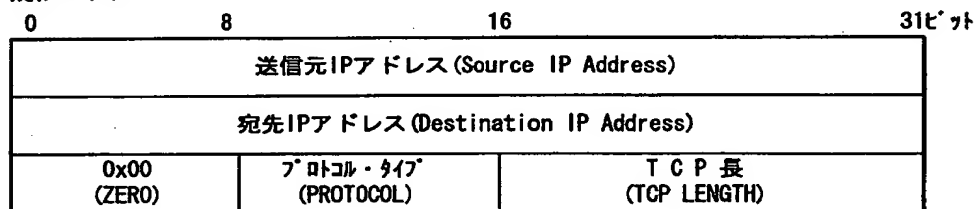
ビッグパケットのヘッダTCP/IPのフォーマット例



(2) コントロールフラグ

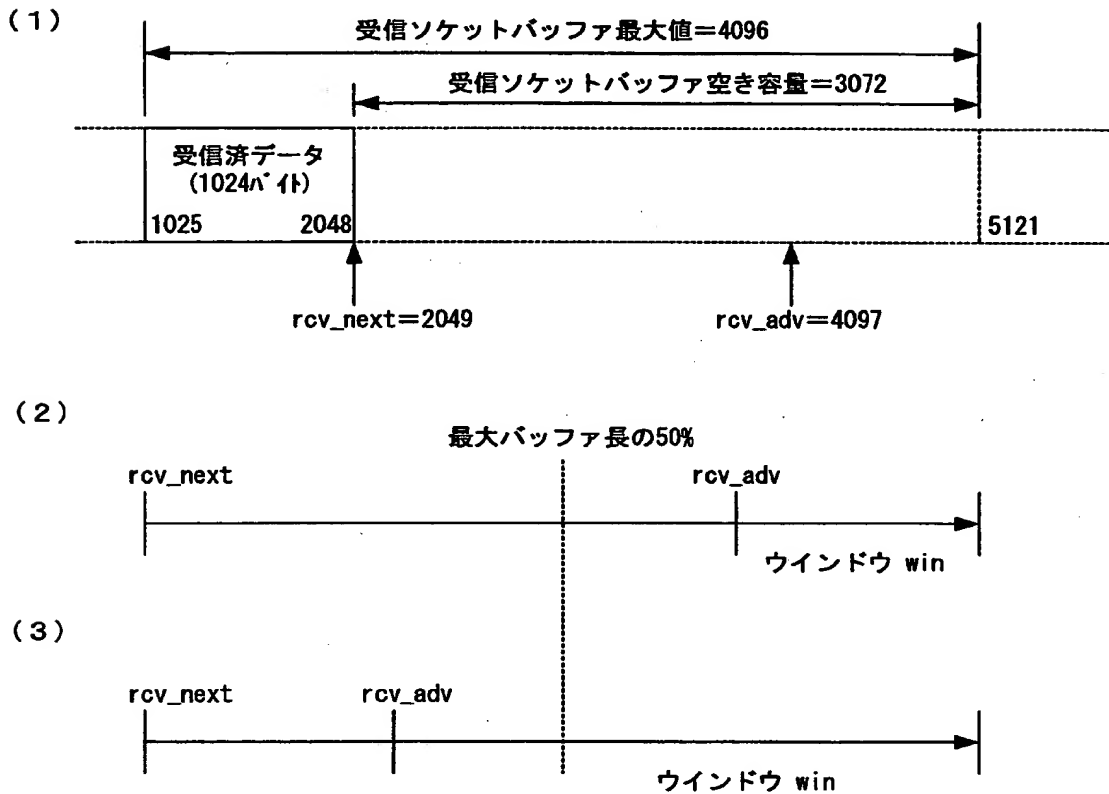


(3) 疑似ヘッダ



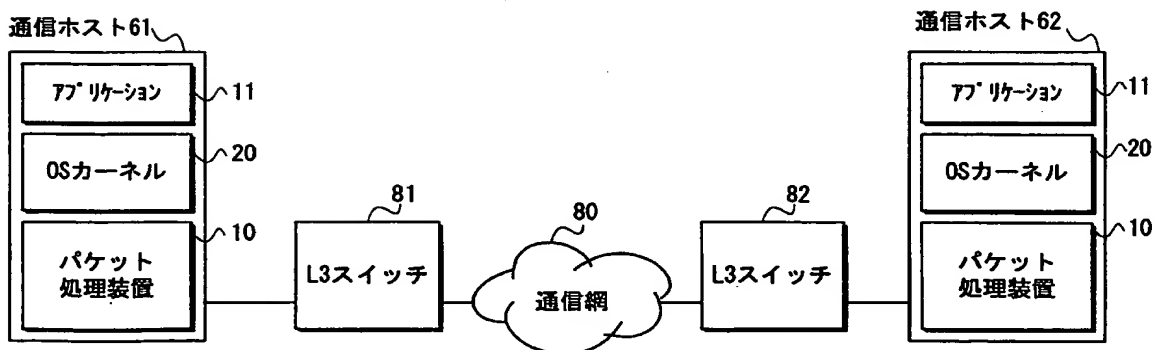
【図 1 7】

ACKパケットの送信タイミング



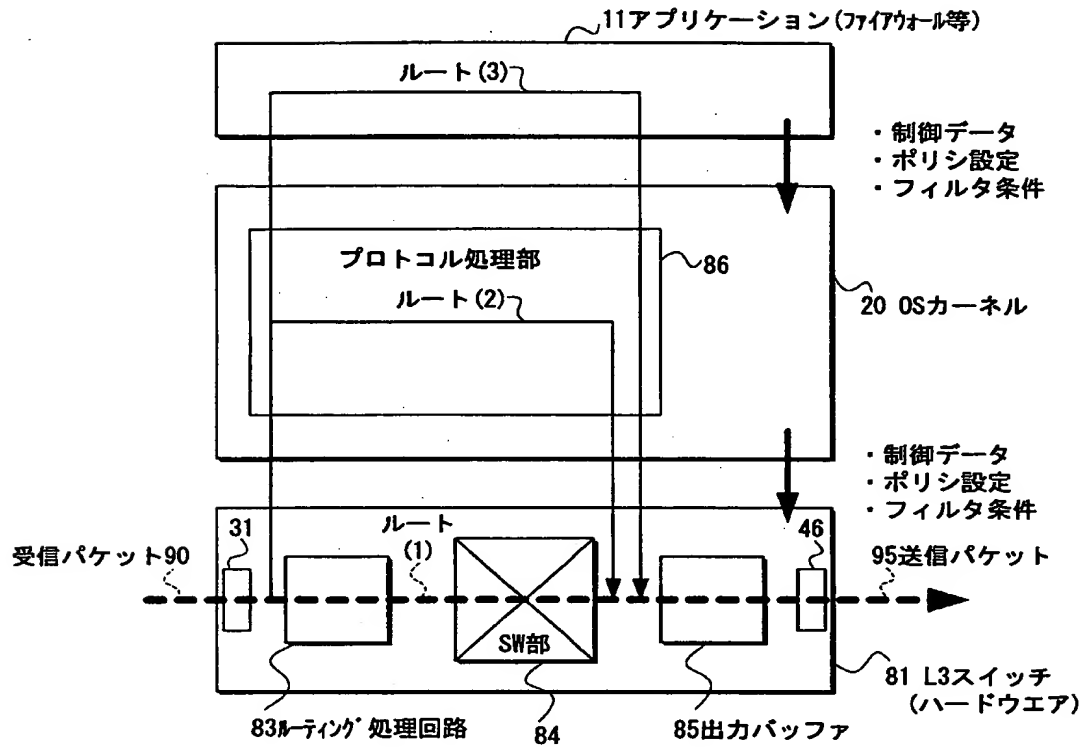
【図 1 8】

本発明の実施例（5）



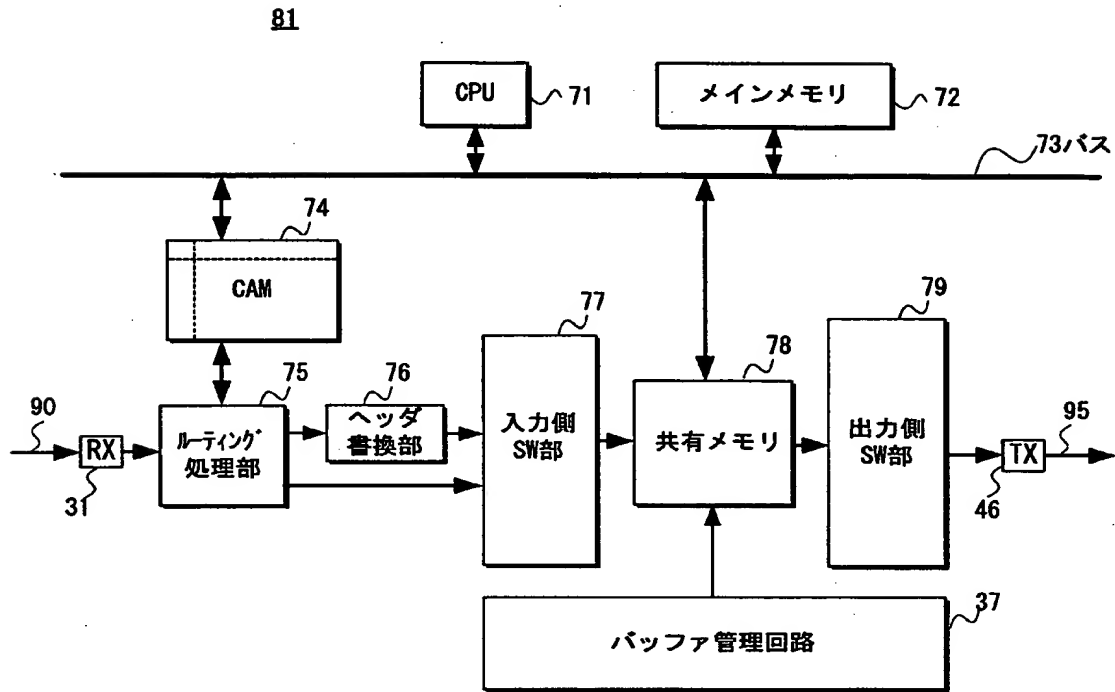
【図 19】

従来のL3スイッチ



【図 20】

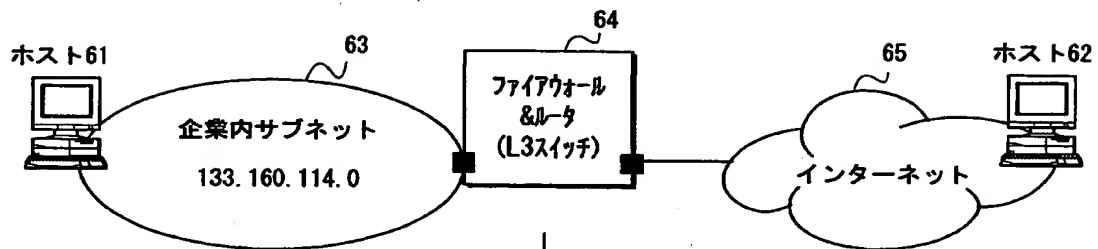
従来の L3 スイッチの構成例



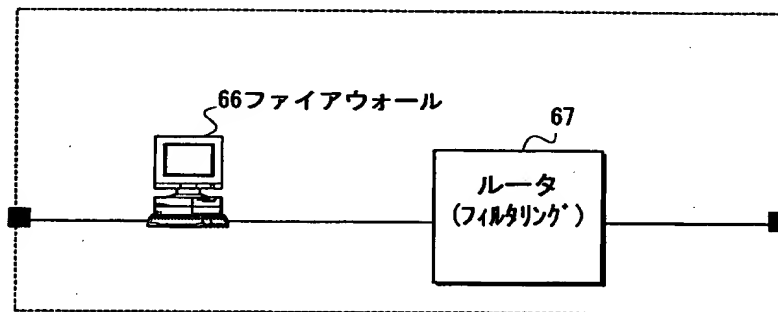
【図 21】

一般的なL3スイッチの応用例

(1)



(2)



【書類名】 要約書

【要約】

【課題】 階層化された通信プロトコルを持つパケット処理装置に関し、各階層のパケット処理が他階層の処理に影響を与えることなく、パケット処理を高速化する。

【解決手段】 空き容量通知部102が上位層100の受信バッファ101の空き容量93を通知し、蓄積条件判定部303が、ビッグパケット92のサイズを空き容量93に基づいて決定し、再構成バッファ処理部301が複数の受信パケット91を1つのビッグパケット92に再構成して受信バッファ101へ送信するように構成する。該空き容量通知部102として上位層100からの逆方向パケット内の情報に基づき該空き容量93を検出する下位層300に配置した逆方向パケット内情報読出回路を用いることができる。さらに、上位層100として、アプリケーション層を用い、ビッグパケット92をトランスポート層のバッファを介さずに、直接、受信バッファ101に送信してもよい。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号

[000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社